Exploratory Database Visualisation

The Application & Assessment of Data and Dimensionality Reduction

Philip James Barrett

Doctor of Philosophy

The University of Aston in Birmingham

September 1995

THE UNIVERSITY OF ASTON IN BIRMINGHAM

Exploratory Database Visualisation

The Application & Assessment of Data and Dimensionality Reduction

Philip James Barrett
Doctor of Philosophy
1995

Thesis summary

This thesis describes the development of a complete data visualisation system for large tabular databases, such as those commonly found in a business environment.

A state-of-the-art 'cyberspace cell' data visualisation technique was investigated and a powerful visualisation system using it was implemented. Although allowing databases to be explored and conclusions drawn, it had several drawbacks, the majority of which were due to the three-dimensional nature of the visualisation.

A novel two-dimensional generic visualisation system, known as MADEN, was then developed and implemented, based upon a 2-D matrix of 'density plots'. MADEN allows an entire high-dimensional database to be visualised in one window, while permitting close analysis in 'enlargement' windows. Selections of records can be made and examined, and dependencies between fields can be investigated in detail.

MADEN was used as a tool for investigating and assessing many data processing algorithms, firstly data-reducing (clustering) methods, then dimensionality-reducing techniques. These included a new 'directed' form of principal components analysis, several novel applications of artificial neural networks, and discriminant analysis techniques which illustrated how groups within a database can be separated.

To illustrate the power of the system, MADEN was used to explore customer databases from two financial institutions, resulting in a number of discoveries which would be of interest to a marketing manager. Finally, the database of results from the 1992 UK Research Assessment Exercise was analysed. Using MADEN allowed both universities and disciplines to be graphically compared, and supplied some startling revelations, including empirical evidence of the 'Oxbridge factor'.

Additional keywords:
Cyberspace, neural networks, Kohonen map, directed PCA, RAE92.

Acknowledgements

The work described in this thesis was funded by the Engineering and Physical Science Research Council (formerly the Science and Engineering Research Council) and by Recognition Systems (formerly Recognition Research).

I am indebted to the James Watt Memorial Foundation, whose decision to award me the James Watt Research Studentship Supplement allowed me to attend two international conferences and kept my bank manager happy.

I must also express my thanks to my colleagues in the Aston University Neural Computing Research Group, who didn't seem to mind my research having little to do with neural computing, and nothing at all to do with Bayes' Theorem. In particular:

- Mike Tipping for handling the nitty-gritty bits of C++ templating and assisting with my never-ending requests for peculiar features
- David Lowe for assistance, suggestions and constructive criticism
- Chris Williams and Ian Nabney for supplying algorithms and code
- Chris Bishop for letting me talk him into buying an Apple colour printer
- and David Bounds, my supervisor, for pointing me in the right direction.

Finally, enormous thanks are due to my fiancée Jenny, who has had to put up with the nightmare of organising our wedding while I was writing my thesis, and without whom it really would have been a nightmare.

Copyright

List of Contents

Thesis Summary 2
Acknowledgements 3
Copyright 3
List of Contents 4
List of Figures 12
List of Colour Plates 15

1 Introduction 17
1.1 Introduction 18
1.2 Database Characteristics 19
1.2.1 Large size 19
1.2.2 Heterogeneous data types 19
1.2.3 Lack of integrity 19
1.2.4 Example databases 20
1.2.4.1 Mail database 20
1.2.4.2 Finance database 20
1.2.4.3 RAE database 21
1.3 Thesis Structure 22
1.3.1 Graphical presentation 22
1.3.2 Interaction 23
1.3.2.1 Cyberspace 23
1.3.2.2 MADEN 23
1.3.3 Data processing
1.3.3.1 Data reduction 24
1.3.3.2 Dimensionality reduction 24

2 Data Visualisation 25
2.1 Introduction 26
2.2 Why Visualise? 27
2.2.1 Optical processing 27
2.2.2 Data compression 28
2.2.3 Understanding and 'topsight' 28
2.3 Basic Visualisation Techniques 30
2.3.1 Introduction 30
2.3.1.1 Continuous variables 30
2.3.1.2 Discrete variables 30
2.3.1.3 Indices 30
2.3.2 One continuous variable, one index 31
2.3.3 One continuous variable, two indices 31
2.3.4 Two continuous, one discrete variables 33
2.3.5 Three continuous variables 33
2.3.6 Three continuous, one discrete variables 34
2.3.7 Three continuous variables, two indices 34
2.3.8 Four continuous variables, one index 35
2.3.9 Four continuous, one discrete variables 35

2.3.10 *n* continuous variables, one index 36
2.4 Advanced Visualisation Techniques 38
2.4.1 Introduction 38
2.4.2 General-purpose techniques
2.4.2.1 Andrews curves 38
2.4.2.2 Chernoff faces 39
2.4.2.3 Texture 40
2.4.2.4 Colour 40
2.4.2.5 Sonification 40
2.4.2.6 Lenses 41
2.4.2.7 Three-dimensional methods 42
2.4.2.8 Bertin's ideas 42
2.4.3 Specific visualisation systems 43
2.4.3.1 IBM Parallel Visual Explorer 43
2.4.3.2 GIFIC 43
2.4.3.3 VisDB 45
2.4.3.4 TripleSpace 45
2.4.3.5 Telecommunications visualisation 45
2.4.4 The Xerox Information Visualiser 46
2.4.4.1 The perspective wall 46
2.4.4.2 Cone and cam trees 47
2.4.4.3 Architecture 48
2.4.4.4 Conclusions 48
2.4.5 Scatter plot matrix systems 48
2.4.5.1 Carr's binned data plots 48
2.4.5.2 Mihalisin's method 49
2.4.5.3 Boyle's *n*-dimensional visualisation system 49
2.4.5.4 Tweedie's prosection matrix 49
2.5 Conclusions 52
2.5.1 Comparative evaluation 52
2.5.2 Summary 53

3 Visualisation Tools I: The Benediktine Cyberspace Cell 54
3.1 Introduction: What is Cyberspace? 55
3.2 Benedikt's Cyberspace 56
3.2.1 Intrinsic and extrinsic dimensions 56
3.2.2 Unfolding 57
3.2.2.1 Example 57
3.2.3 Benedikt's cell 58
3.3 Implementation 60
3.3.1 Platform 60
3.3.2 Data input 60
3.3.3 Density projection 60
3.3.4 Display 61
3.3.4.1 Cell layout and display 61
3.3.4.2 Wall display 62
3.3.4.3 Axis labels 64

3.3.4.4 Initial choice of fields 65
3.3.5 Overlays 65
3.3.6 Axis selection 68
3.3.7 Vehicle movement 68
3.3.8 Vehicle movement 69
3.3.9 The probe 71
3.3.9.1 Probe size 71
3.3.9.2 Probe display 72
3.3.9.3 Probe control 74
3.3.10 Dependent walls 75
3.3.11 Subspaces 76
3.4 Use with Real Data 78
3.4.1 Mail database 78
3.4.2 RAE database 85
3.5 Conclusions
3.5.1 Concept 87
3.5.2 Implementation 87
3.5.3 Practical applications 89

4 Visualisation Tools II: MADEN 90
4.1 Concept 91
4.1.1 Transformation to a two-dimensional matrix 91
4.1.2 Extension to *n* dimensions 93
4.1.3 Alternative layouts 93
4.1.4 'MADEN' 94
4.2 Implementation
4.2.1 Code structure 95
4.2.2 Overview display 95
4.2.3 Density plots 97
4.2.3.1 Density matrix creation 97
4.2.3.2 Density plot display 98
4.2.4 Axis labelling 100
4.2.5 Axis selection and ordering 100
4.2.6 Initial field choice 102
4.2.7 Overlays 103
4.2.8 Enlargements 105
4.2.8.1 Axis modification 107
4.2.9 Selection 107
4.2.9.1 Selection display 107
4.2.9.2 Selection control 110
4.2.10 Dependent enlargements 111
4.2.11 Clipping 112
4.2.12 Data saving 112
4.2.13 Window deletion 112
4.2.14 Footers 113
4.3 Use with Real Data
4.3.1 Mail database 114

4.3.1.1 Use of enlargements 114
4.3.1.2 Use of overlays 116
4.3.1.3 Use of dependent enlargements 117
4.3.2 Finance database 119
4.3.3 RAE database
4.3.3.1 Difficulties 121
4.3.3.2 Alternate overlay colour scale 121
4.3.3.3 Field division 123
4.3.3.4 'Serial number' fields 124
4.3.3.5 Highlighting 124
4.4 Conclusions
4.4.1 Concept 127
4.4.2 Implementation 127
4.4.3 Evaluation 129

5 Data Reduction: Clustering 130
5.1 Introduction 131
5.1.1 Segmentation 131
5.1.2 Requirements 132
5.1.2.1 Distance measure 132
5.1.2.2 Clustering algorithm 132
5.1.2.3 Display method 133
5.2 Distance Measures
5.2.1 Notation 134
5.2.2 Distance metrics 134
5.2.2.1 Euclidean metric 135
5.2.2.2 City block metric 135
5.2.2.3 Minkowski metrics 135
5.2.3 Differences between categorical fields 136
5.2.4 Choice of measure 136
5.3 Sequential Leader Algorithm 136
5.3.1 Algorithm 137
5.3.2 Performance 137
5.3.3 Assessment
5.3.3.1 Model accuracy 138
5.3.3.2 Speed 138
5.3.3.3 Number of clusters 138
5.4 FASTCLUS 139
5.4.1 Algorithm 139
5.4.2 Results 139
5.4.3 Assessment
5.4.3.1 Model accuracy 140
5.4.3.2 Speed 140
5.4.3.3 Number of clusters 140
5.5 Mixture Models 141
5.5.1 Algorithm 141
5.5.1.1 Initialisation 141

5.5.1.2 Expectation step 142
5.5.1.3 Maximisation step 142
5.5.1.4 Likelihood 143
5.5.2 Results 143
5.5.3 Assessment
5.5.3.1 Model accuracy 144
5.5.3.2 Speed 144
5.5.3.3 Number of clusters 144
5.5.4 Alternative approaches 144
5.6 Choice of Algorithm 145
5.6.1 Implementation 145
5.7 Display Techniques 146
5.7.1 Centres 146
5.7.2 Ellipses 147
5.7.3 Multiple ellipses 148
5.7.4 Density plots 149
5.7.5 Conclusions 150
5.8 The Kohonen Self-Organising Map
5.8.1 Introduction 151
5.8.2 Structure 151
5.8.3 Training algorithm 151
5.8.3.1 Update equations 153
5.8.4 Training process 154
5.8.5 Implementation 155
5.9 Use with Real Data
5.9.1 Mail database
5.9.1.1 Mixture model 156
5.9.1.2 Kohonen map 158
5.9.2 Finance database
5.9.2.1 Mixture model 163
5.9.2.2 Kohonen map 164
5.9.3 RAE database
5.9.3.1 Mixture model 168
5.9.3.2 Kohonen map 168
5.10 Conclusions 172
5.10.1 Summary 173

6 Dimensionality Reduction I: Linear Methods 174
6.1 Introduction 175
6.1.1 Trivial field reduction 175
6.1.2 Dimensionality reduction 175
6.1.3 Linear projection 176
6.2 Principal Component Analysis 177
6.2.1 Procedure 177
6.3 Factor Analysis 179
6.3.1 Principal factor analysis 179
6.3.1.1 Explanation 180

6.3.2 Factor rotation 181
6.3.3 Choice of number of factors 181
6.3.4 Projection 182
6.4 Directed Principal Component Analysis 183
6.5 Projection Pursuit 184
6.5.1 Introduction 184
6.5.2 Algorithm 184
6.5.3 Projection indices 184
6.5.3.1 Common elements 185
6.5.3.2 Holes and central mass indices 185
6.5.3.3 Skew index 186
6.5.4 Implementation 186
6.6 Linear Discriminant Analysis 187
6.6.1 Procedure 187
6.7 Implementation
6.7.1 Initiation 189
6.7.2 Pre-processing 189
6.7.3 Choice of number of factors 189
6.7.4 Choice of active PP fields 190
6.7.5 Display of axes 190
6.7.6 Factor post-processing 191
6.7.7 Projection 191
6.8 Use with Real Data 192
6.8.1 Mail database
6.8.1.1 Principal component analysis 192
6.8.1.2 Factor analysis 194
6.8.1.3 Directed principal component analysis 196
6.8.1.4 Projection pursuit 198
6.8.1.5 Linear discriminant analysis 200
6.8.2 Finance database
6.8.2.1 Principal component analysis 203
6.8.2.2 Directed principal component analysis 205
6.8.2.3 Projection pursuit 207
6.8.2.4 Linear discriminant analysis 208
6.8.3 RAE database
6.8.3.1 Factor Analysis 209
6.8.3.2 Directed principal component analysis 213
6.8.3.3 Projection pursuit 215
6.8.3.4 Linear discriminant analysis 218
6.8.3.5 Combined approach 224
6.9 Conclusions 227

7 Dimensionality Reduction II: Non-linear Methods 229
7.1 Introduction 230
7.2 Traditional Methods
7.2.1 Sammon's non-linear mapping 231
7.2.1.1 Application of neural networks 232

7.2.2 Multidimensional scaling 232
7.2.3 Principal curves and surfaces 232
7.2.4 Conclusions 233
7.3 Kohonen Self-organising Map Revisited
7.3.1 Concept 234
7.3.2 Implementation 234
7.4 Multi-layer Perceptron Hidden Layer
7.4.1 Introduction to MLPs 236
7.4.1.1 Choice of activation function 236
7.4.2 Application of an MLP to dimensionality reduction 237
7.4.3 Implementation 238
7.5 Multi-layer Perceptron Autoencoder
7.5.1 Introduction 239
7.5.2 Architecture 239
7.5.3 Implementation 240
7.6 Use with Real Data
7.6.1 Mail database
7.6.1.1 Kohonen mapping 241
7.6.1.2 Hidden layer 244
7.6.1.3 Autoencoder 247
7.6.2 Finance database
7.6.2.1 Kohonen mapping 252
7.6.2.2 Hidden layer 253
7.6.2.3 Autoencoder 255
7.6.3 RAE database 256
7.6.3.1 Kohonen mapping 256
7.6.3.2 Hidden layer 263
7.6.3.3 Autoencoder 265
7.7 Conclusions 266

8 Conclusions 267
8.1 Conclusions 268
8.1.1 The Benediktine cyberspace cell 268
8.1.2 MADEN 268
8.1.3 Data reduction 268
8.1.4 Linear dimensionality reduction 269
8.1.5 Non-linear dimensionality reduction 269
8.1.6 Summary 270
8.2 Summary of Results 271
8.2.1 Mail database 271
8.2.2 Finance database 272
8.2.3 RAE database 272
8.3 Further Work 274

List of References 275

Appendix: Database Details 284
A.1 Introduction 284
A.2 Mail Database 284
A.2.1 Field details 284
A.3 Finance Database 285
A.3.1 Field details 285
A.4 RAE Database
A.4.1 References 286
A.4.2 Pre-processing 286
A.4.3 Field details 287
A.4.4 Institution codes 289
A.4.5 Unit of assessment codes 291

List of Figures

1.1 Research Assessment Exercise rating scale 21

2.1 Simple bar chart 31
2.2 3-D bar chart 31
2.3 Contour plot 32
2.4 3-D wireframe surface plot 32
2.5 3-D shaded surface plot 32
2.6 2-D scatter plot 33
2.7 Percentage ternary plot 33
2.8 *x-y* shaded contour plot 33
2.9 Bubble plot 34
2.10 3-D scatter plot 34
2.11 3-D scatter line plot 34
2.12 Box plot 35
2.13 2-D vector plot 35
2.14 Line graph 36
2.15 3-D ribbon plot 36
2.16 Line graph with error bars 36
2.17 Stepped line graph 36
2.18 Bar chart 36
2.19 Stacked bar chart 36
2.20 Bar/line overlay chart 37
2.21 Pie chart stack 37
2.22 Spider plot 37
2.23 An Andrews plot, showing 150 Andrews curves 39
2.24 Comparison of features of six visualisation techniques 53

3.1 Initial view of the cell (using the mail database) 62
3.2 Example of wall displays 63
3.3 Solutions to high-density areas along the axes 64
3.4 Walls control panel 68
3.5 Axis selection popup menu (truncated) 69
3.6 View after moving and turning the vehicle 70
3.7 Properties control panel 71
3.8 Evolution of the probe display 74
3.9 Subspace information window. 75
3.10 Display showing a subspace 77
3.11 A starting point for exploration of the mail database 78
3.12 Subspace information window showing details of the selection of high respondents 81
3.13 Wall showing number of atm withdrawals (*x* axis) against maximum balance (*y* axis) 82
3.14 Wall showing number of ATM withdrawals (*x* axis) against maximum balance (*y* axis) dependent on selection of customer age (*z* axis, not seen) 83

3.15 Walls showing number of ATM withdrawals (x axis) against maximum balance (y axis) for customer age zero, i.e. where the age is unknown and for non-zero customer age, i.e. where the age is known 85
3.16 Wall showing `inpost` against `pub1` from the RAE database 85
3.17 Wall showing `sel_staff` against `rating` from the RAE database 86

4.1 Transformation from the 3-D Benediktine cell to a 2-D matrix 92
4.2 The four possible matrix layouts 93
4.3 Continuous/continuous density plot 98
4.4 Continuous/discrete density plots 98
4.5 Discrete/discrete density plots 99
4.7 An axis popup menu 101
4.8 Displayed axes window for the mail database 101
4.9 Overview overlay menu 103
4.10 Example enlargements 105
4.11 A highly enlarged density plot 106
4.12 A field control window 110
4.13 Left hand footer of the overview window 113
4.14 Right hand footer of the overview window 113
4.15 `Ac_Age–Age` enlargement 114
4.16 `Home–Age` enlargement 115
4.17 `Home–Dcard` enlargement 115
4.18 Overview of the entire finance database 119
4.19 Enlargements showing the effect of the divide operation 123
4.20 Visualisation evaluation chart, including MADEN 129

5.1 Performance of the sequential leader algorithm on real data 138
5.2 Performance of the FASTCLUS algorithm on real data 140
5.3 Enlarged view of the projection before clustering 146
5.4 Cluster display showing centres 147
5.5 Cluster display showing ellipses at the standard deviation of the clusters 147
5.6 Cluster display showing filled ellipses 148
5.7 Cluster display showing three ellipses per cluster 148
5.8 Cluster displays showing multiple ellipses 149
5.9 Cluster displays showing density plots 150
5.10 Structure of the Kohonen map, showing the coordinate system 152
5.11 The Kohonen map, showing the winning node c at (3,2) and neighbourhood radii on other nodes 154
5.12 Overview of five fields of the clustered mail database 156
5.13 Example enlargement from the fifty clusters of the full mail database 157
5.14 Comparison of original and clustered enlargements from the mail database 157
5.15 Averaged error during training of Kohonen map on the mail database 158
5.16 Overview of six components of the Kohonen weight vectors from the mail database 159
5.17 Demonstration of the modelling power of the Kohonen map 159
5.18 Comparison of original and clustered enlargements from the finance database 163

5.19 Overview of six components of the Kohonen weight vectors from the finance database 164
5.20 Overview of six fields of the finance database 165

6.1 First six principal components of the mail database 192
6.2 Projection of the mail database onto its first six principal components 194
6.3 Explaining power of one to six factors of the mail database 194
6.4 Three factors of the mail database 195
6.5 Overview of the (estimated) factor scores of the mail database, using three rotated factors 196
6.6 First four DPCA axes of the mail database 197
6.7 Result of PP on the mail database using central mass index 199
6.8 Result of PP on the mail database using skew index 199
6.9 Linear discriminant axis of the mail database 201
6.10 Canonical variate of the mail database 201
6.11 First five principal components of the finance database 203
6.12 Overview of finance database projected onto five principal components 204
6.13 First four DPCA axes of the finance database 205
6.14 Result of PP on the finance database using skew index 207
6.15 Linear discriminant axis of the finance database 208
6.16 One principal factor of the RAE database, and the plot of its factor scores against `rating` 209
6.17 Relationships between standardised variables and `rating` 212
6.18 First three DPCA axes of the RAE database 213
6.19 PP axes of the RAE database using skew index 215
6.20 PP projections of the RAE database using skew index 216
6.21 Results of PP on the people-related standardised RAE database using skew index 217
6.22 Two most significant linear discriminant axes of the RAE database 218
6.23 Two canonical variates of the RAE database plotted against `rating` 219
6.24 RAE canonical variates, dependent upon unit of assessment 221
6.25 RAE canonical variates, dependent upon institution 222
6.26 Two most significant linear discriminant axes of the standardised RAE database 224

7.1 Enlargement of the Kohonen map coordinates showing data density 235
7.2 Multi-layer perceptron architecture (for RAE database) 237
7.3 Autoencoder architecture 239
7.4 Three-dimensional autoencoding of the mail database 247
7.5 Three-dimensional autoencoding of the finance database 255
7.6 RAE Kohonen mapping, dependent upon unit of assessment 257
7.7 RAE Kohonen mapping, dependent upon institution 258
7.8 Results of training a three-node bottleneck MLP autoencoder on the RAE database 265

List of Colour Plates

2.1 Example of the use of a table lens 41
2.2 PVE being used to explore the money markets 43
2.3 GIFIC display showing two hearts 44
2.4 GIFIC display showing 100 combat troops 44
2.5 Visual Recall grid view, based on the perspective wall 46
2.6 Visual Recall tree view, similar to a cam tree 47
2.7 Boyle's *n*-dimensional visualisation system 49
2.8 Tweedie's explanation of prosection 50
2.9 The prosection matrix 51
2.10 Adjusting the tolerance on the prosection matrix 51

3.1 Appearance of the overlay on a continuous/continuous wall 66
3.2 Appearance of the overlay on a continuous/discrete wall 67
3.3 Appearance of the overlay on a discrete/discrete wall, with the overlay field the same as the *x* axis field 67
3.4 A view of the mail database with the `Response` field overlaid 79
3.5 A different view of the mail database, with a selection of likely responders 80
3.6 The subspace opened from the selection shown in plate 3.5 81

4.1 Overview display showing the entire mail database 95
4.2 Overview display showing five fields of the finance database 96
4.3 Example of an overview with an overlay 104
4.4 Two enlargements of the same density plot with different overlays 106
4.5 Overview showing a selection made on three axes 109
4.6 Enlargement window showing a selection 109
4.7 `No_Wdraw–Ac_Turn` enlargement with `Response` overlaid 116
4.8 `Home–Marital` enlargement with `Response` overlay, dependent on selection made on `Age` 117
4.9 Two enlargements from the finance database 120
4.10 8 fields of the RAE database with `rating` overlaid 121
4.11 8 fields of the RAE database with `rating` overlaid using new colour scale 122
4.12 Enlargement showing outliers which will be highlighted 125
4.13 Overview showing a highlighted data record 126

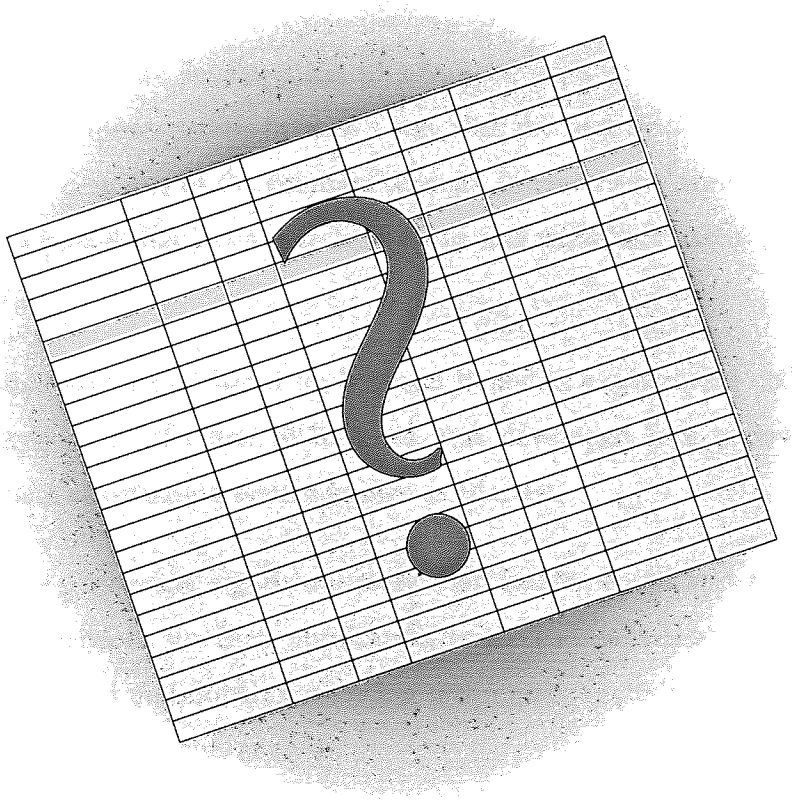
5.1 Kohonen weight vector distributions from the mail database 161
5.2 Kohonen weight vector distribution and separation measure from the mail database 162
5.3 Kohonen node separation measure from the finance database 166
5.4 Kohonen weight vector distributions from the finance database 167
5.5 Kohonen weight vector distributions and separation measure from the RAE database 169
5.6 Kohonen weight vector distributions from the RAE database (three left columns removed) 170

6.1 Axis choice window following PCA on the mail database 191
6.2 Mail database projected onto four DPCA axes, with `Response` overlaid 198

6.3 Results of PP on the mail database, with Response overlaid 200
6.4 Canonical variate of the mail database, with Response overlaid..... 202
6.5 Finance database projected onto four DPCA axes, with response overlaid. 206
6.6 Result of PP on the finance database using skew index, with response overlaid 207
6.7 Canonical variate of the finance database with response overlaid 208
6.8 Factor scores of the first factor of the RAE database plotted against standardised sel_staf, with rating overlaid. 210
6.9 RAE database projected onto three DPCA axes, with rating overlaid 214
6.10 PP projections of the RAE database using skew index, with rating overlaid 216
6.11 Result of PP on the people-related standardised RAE database using skew index, with rating overlaid 217
6.12 Two canonical variates of the RAE database with rating overlaid..... 220
6.13 Canonical variates of the standardised RAE database with rating overlaid .225

7.1 Kohonen mapping of the mail database with overlays 242
7.2 Kohonen mapping of the mail database with overlays 243
7.3 Results of training a three-node bottleneck MLP on the mail database, with predicted response overlaid..... 244
7.4 Results of training a three-node bottleneck MLP on the mail database, with actual Response overlaid 245
7.5 Canonical variate of the hidden layer activations, with Response overlaid 246
7.6 Three-dimensional autoencoding of the mail database, with Response overlaid 248
7.7 Three-dimensional autoencoding of the mail database, with Home:R overlaid 249
7.8 Three-dimensional autoencoding of the mail database, with Marital:M overlaid 250
7.9 Three-dimensional autoencoding of the mail database, with Dcard overlaid 251
7.10 Kohonen mapping of the finance database with overlays 252
7.11 Results of training a three-node bottleneck MLP on the finance database, with predicted response overlaid..... 253
7.12 Results of training a three-node bottleneck MLP on the finance database, with actual response overlaid 254
7.13 Standardised RAE Kohonen mapping, dependent on unit of assessment..... 260
7.14 Standardised RAE Kohonen mapping, dependent upon institution 261
7.15 Results of training a three-node bottleneck MLP on the RAE database, with predicted rating overlaid..... 263
7.16 Results of training a three-node bottleneck MLP on the RAE database, with actual Rating overlaid 264

Chapter 1



Introduction

Well, this bit which I am writing, called Introduction, is really the er-h'r'm *of the book, and I have put it in, partly so as not to take you by surprise, and partly because I can't do without it now. There are some very clever writers who say that it is quite easy not to have an* er-h'r'm, *but I don't agree with them. I think it is much easier not to have the rest of the book.* [Milne, 1927]

1.1 Introduction

There is a wealth of information hidden in every database, but traditionally the users of such databases have lacked the tools to extract it. Existing database applications will allow one record to be viewed in detail, permit the extraction of all the records which meet a given set of criteria, and maybe offer some simple statistical analyses on a set of records, but there are very few tools which allow the entire database to be presented in any graphical form more complex than a pie chart.

This thesis describes the development of advanced tools to allow a database to be displayed on a computer screen in such a way that the user of the software can gain insights into the contents of the database through interactively 'exploring' it in some fashion. It was hoped that a commercially viable and genuinely useful product could be developed.

An initial visualisation tool based on Michael Benedikt's cyberspace cell [Benedikt, 1991B], believed to be the first of its kind, was developed into a powerful novel two-dimensional visualisation system known as MADEN.

Numerous data processing algorithms were then implemented and assessed within the MADEN framework for their application to data visualisation. The algorithms included data- and dimensionality-reducing techniques, including a novel 'directed' form of principal components analysis.

Investigations were made into whether artificial neural networks had anything new to offer to the field of visualisation, and indeed whether the visualisation techniques developed could have applications in visualising the internal behaviour of the networks.

Each data processing technique was assessed by attempting to extract more and more information from three sample databases, including the database of results from the 1992 Research Assessment Exercise.

This research was carried out under Aston's Interdisciplinary Higher Degree scheme, requiring links to an external company, in this case Recognition Systems (formerly Recognition Research). This is a small company based on the Aston University Science Park, which specialises in the business applications of neural network technology. As such, a lot of its work involves business databases, in the contexts of predicting customer behaviour and imputing missing data.

1.2 Database Characteristics

The tabular databases which are visually explored in this thesis are typically business databases, and have three characteristic features as detailed below.

1.2.1 Large size

The databases are large, both in terms of the number of records (e.g. one per customer) and the number of fields in each record (the number of pieces of information relating to that customer). A typical database might have ten thousand records, each with thirty fields.

The large sizes involved cause great problems for data analysis and visualisation. Carr [Carr, 1991] refers to 'the monumental challenge of developing exploratory analysis methods for large data sets'.

1.2.2 Heterogeneous data types

The fields are generally heterogeneous, including some if not all of the following:

- continuous numeric (e.g. minimum bank balance=£2.41)
- integral numeric (e.g. number of children=3)
- ordered categorical (e.g. socioeconomic group=C2)
- unordered categorical (e.g. marital status=Single)
- binary (e.g. has credit card=TRUE)
- textual (e.g. name="J Major").

1.2.3 Lack of integrity

Missing data is a major problem for data analysis [Little & Rubin, 1987], particularly in business databases, where there is often a high proportion of records with missing values in certain fields – such as where the customer's age is unknown – which are typically coded using a value such as zero, minus one or 'U'.

1.2.4 Example databases

Three databases were obtained to develop and test the visualisation tools described in this thesis. Two were customer databases from UK financial institutions, and were supplied by Recognition Systems, the third was the public domain database of results from the 1992 Universities Research Assessment Exercise.

Detailed descriptions of the example databases are contained in the appendix. Below is an introduction to each database.

1.2.4.1 Mail database

This database originated with a financial institution, and is the result of simulations based on genuine customer responses to mailed advertisements for life insurance. The database contains 10,000 records of 20 fields – 19 fields of information about each customer, and a response field indicating whether that customer responded positively to the mail shot. Almost exactly half the customers in the database made positive responses.

In visualising this database, it is hoped to learn something about the structure of the database (e.g. whether there are distinct groups or 'segments' of customers), and to identify which characteristics are shared by customers who are likely to respond to the mail shot. In this way, the cost of sending another mail shot can be reduced by only mailing the customers with the same characteristics.

1.2.4.2 Finance database

This database is very similar to the mail database, in that it contains customer information from a financial institution, and is of a similar size (10,339 records each with 22 fields). One of the fields is a binary response, in this case indicating whether the customer took a desired action. Unlike the mail database, the finance data came from actual customer records, and to protect client confidentiality, Recognition Systems added noise to the data, changed the scales of various fields, and removed the field names.

The primary aim in visualising this database is to identify customers with a positive response. It may also be possible to identify structures in the database, but since the original field names are not known, such discoveries cannot be interpreted.

1.2.4.3 RAE database

Every three years since 1986, the UK Universities Funding Council has undertaken a Research Assessment Exercise (RAE). This exercise consists of gathering information from every academic research institution in the UK – including the number of publications in various categories, details of research grants, numbers of staff and research students, etc. A peer review process then uses this information, together with subjective information (such as the 'quality' of publications and current research), to award each university department a 'research rating' – an integer from one to five, as shown in figure 1.1.

Rating	Description of research quality
1	National level of excellence in none, or virtually none, of the subareas of activity
2	National level of excellence in up to half of the subareas of activity
3	National level of excellence in a majority of the subareas of activity; or to international level in some
4	National level of excellence in virtually all subareas of activity, possibly showing some evidence of international excellence; or to international level in some and at least national level in a majority
5	International level of excellence in some subareas of activity and national level in virtually all others

Figure 1.1 – Research Assessment Exercise rating scale [Johnes *et al*, 1993]

The database of results from the 1992 exercise (following pre-processing as detailed in the appendix) has most of the characteristics defined earlier. It has 2706 records (one for each university department), each with 84 fields, one of which is the research rating awarded by the Council.

One of the aims of visualising this data is to identify which characteristics are shared by those departments which were awarded the same rating. In particular, it would be interesting to discover whether there is a 'recipe for a five', or to compare disciplines, to see whether the criteria for a high rating in a Law department were the same as those in Nursing, for example. A third possibility is to search for departments which are very different (in some way) from the others in the database, and see how this affects their research rating.

1.3 Thesis Structure

> **visualize** /ˈvɪzjʊəˌlaɪz, ˈvɪʒj-/ *v.tr.* (also **-ise**) 1 make visible esp. to one's mind (a thing not visible to the eye). 2 make visible to the eye. **visualizable** *adj.* **visualization** /-ˈzeɪʃ(ə)n/ *n.* [Allen, 1990]

This is the definition of 'visualisation' supplied by *The Concise Oxford Dictionary*. A generally-accepted current definition of 'data visualisation' in the scientific community is:

- *Presenting data in a graphical form*
- *Allowing the user to interactively 'explore' the data*
- *Processing the data to aid exploration*

This definition lays out the structure of this thesis.

1.3.1 Graphical presentation

> *There are numerous time-tested scientific visualisation techniques for producing pretty pictures while avoiding unnecessary illumination of the data.* [Globus & Raible, 1994]

Primarily, visualisation is all about generating pictures. From the simplest histogram to a highly complex 3-D animation in a virtual environment, the supplied data is used to create something graphical.

Some kinds of data displayed on a computer screen or printout can be understood at a glance: line graphs of exchange rates, cross-sections of the sea bed, simple weather maps, traffic flow in a city centre, etc. Other kinds are not so easily visualised: the detailed state of the world's stock markets, measurements from hundreds of sensors on a satellite, the performance of a model in a wind tunnel, the acoustic qualities of an auditorium, the demographic make-up of a section of the community, the database of customers of a financial institution etc.

For these sorts of data, the visualisation techniques need to be a little more advanced – maybe involving the use of colour, texture, movement, 3-D images, even sound. Any image which is generated by reference to the data in question may rightly be described as a visualisation of that data.

Of course, some techniques are better than others for highlighting certain features of the data – showing trends over time, emphasising outlying data points, picking out clusters etc. Many may be totally inappropriate for a given dataset, yielding merely a seemingly random or uniform display, but by choosing a suitable one, features of the data which were previously hidden will be made visible.

An overview of some basic visualisation techniques, and an exploration of research into more advanced visualisation will be covered in chapter 2.

1.3.2 Interaction

An ideal visualisation tool will allow the user to interactively 'explore' the database, changing what is being displayed in order to seek out specific information. For example, when looking at the results of a census overlaid on a map, it would be useful to be able to explore the changes in population distribution as various age ranges are selected. If this can be achieved in near-real time, the user could simply move a sliding control on the screen and watch the distribution change accordingly. The (near-) instant feedback of such a system creates a very powerful exploration tool [Jog & Shneiderman, 1994].

1.3.2.1 Cyberspace

Taking data interaction to its extreme results in the concept of 'cyberspace', a virtual environment in which everything seen and heard is a representation of data. Michael Benedikt's idea of the cyberspace 'cell' [Benedikt, 1991B] has applications for visualising large databases, and a full implementation is described in chapter 3.

1.3.2.2 Maden

However, the Benediktine cell as implemented proved to be unsuitable for visualising the example databases. Chapter 4 details the development of a novel visualisation tool, MADEN, which uses a matrix of 'density plots' to visualise multidimensional databases. MADEN is shown to overcome the limitations of the Benediktine system, and to be an ideal test-bed for the data processing operations considered in the following chapters.

1.3.3 Data processing

1.3.3.1 Data reduction

The usefulness of a visualisation system is often limited by its speed of response. When visualising a large database, the slow response may be attributable to the number of data records in the database. If the size of the database can be reduced, the processing time required to update a display showing the data will also be reduced.

Chapter 5 describes the results of experiments with algorithms designed to find a small number of 'clusters' of data records in the database, in the hope that the resulting clusters can then be visualised faster than the original data. The Kohonen self-organising map [Kohonen, 1990] is shown to be a particularly powerful tool.

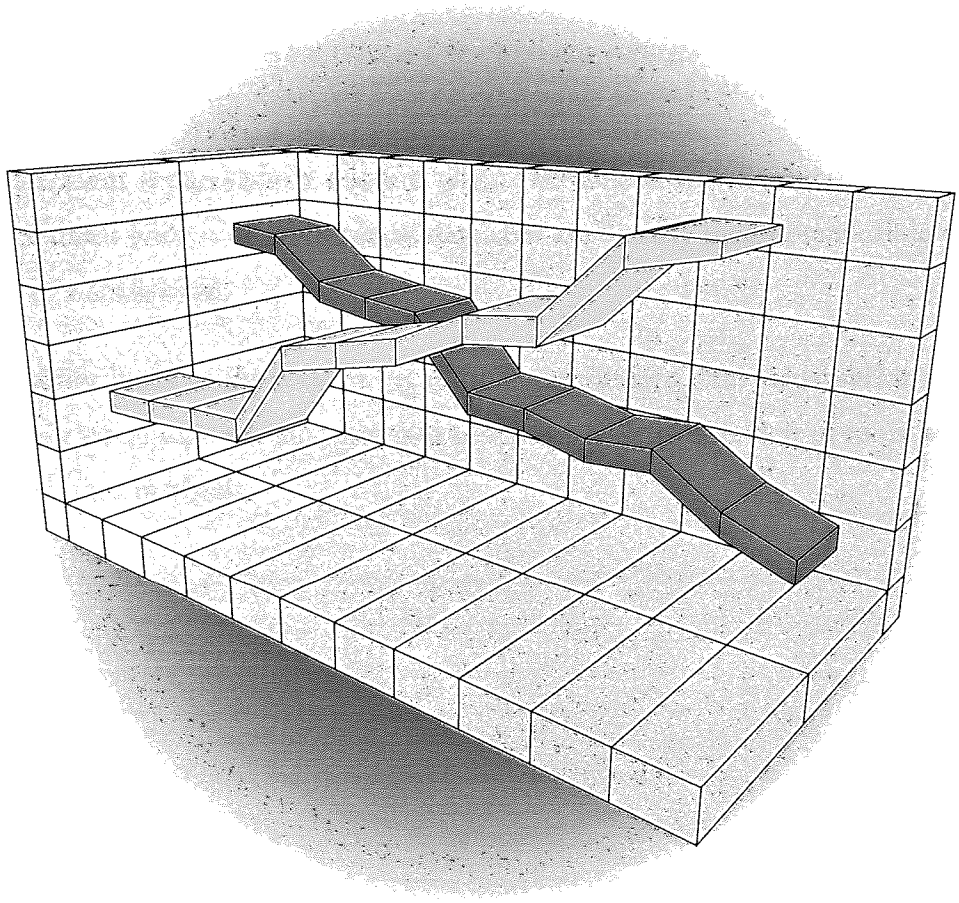
1.3.3.2 Dimensionality reduction

Alternatively, the size of the database may be reduced by reducing the number of dimensions of the data. Dimensionality reduction is of course of particular importance where there is a high number of dimensions in the raw database. If the choice of processing algorithm is made carefully, so that little useful information is discarded, the user should be able to see all the important features of the original dataset by viewing a visualisation of the transformed dataset.

The investigation of dimensionality reduction techniques is begun in chapter 6 which considers linear projection methods, and introduces a novel form of principal components analysis which is 'directed' towards a result which is meaningful in the context of a particular database. Projection pursuit algorithms which attempt to automatically select a particular reduced-dimensionality view of the data which shows something 'interesting' are also assessed.

Chapter 7 moves on to investigate non-linear dimensionality reduction techniques utilising artificial neural networks, including novel applications of two types of multi-layer perceptron networks and another, more powerful, application of the Kohonen self-organising map.

Chapter 2



Data Visualisation

To envision information – and what bright and splendid visions can result – is to work at the intersection of image, word, number, art. The instruments are those of writing and typography, of managing large data sets and statistical analysis, of line and layout and color. [Tufte, 1990]

2.1 Introduction

Walker [Walker *et al*, 1993] states that the term *visualisation* was first popularised in a 1987 special issue of *Computer Graphics* [McCormick *et al*, 1987]. However, what we now know as visualisation has been used for centuries, whenever someone has found that drawing a picture is a good way of conveying some information. Long before the advent of computers, it was recognised that tables of figures were far from ideal:

> *Getting information from a table is like extracting sunlight from a cucumber*
> [Farquhar & Farquhar, 1891]

This statement is particularly relevant when the table is ten thousand lines long and thirty columns wide. Today's business databases are literally incomprehensible without the aid of visualisation.

This chapter looks at the reasons for using computerised data visualisation, outlines some basic visualisation techniques, and concludes with an overview of more advanced and specialised methods.

2.2 Why Visualise?

There are three main justifications for performing the computationally-expensive task of data visualisation: optical processing, data compression, and the quest for understanding.

2.2.1 Optical processing

The primary motivation for visualisation is to exploit the incredible human ability to quickly assimilate, correlate, comprehend and interpret vast amounts of information optically. From birth, as long as we have our eyes open, we are all subjected to visual stimulation. The decades of subconscious training we have undergone enable us to react to particular visual stimuli almost without thinking.

A changing pattern on a computer display can be immediately meaningful to a viewer, if it behaves in a manner which is related to reality. Much of the ease of use of the current window-based user interfaces relies on the basic premise that the windows on the screen are arranged 'on top of one another' and those closer to the user can 'obscure' those behind – which, as our brains know very well, is what happens in the real world.

A key ability of the human eye is its capacity for colour vision, and the use of colour in images can be very powerful. A single pixel in a display of blue which changes colour to red, for example, will be immediately noticed by an observer. If the colours used by a system are carefully chosen so that unusual values 'stand out', a simple display allows a user to easily identify items which deviate from the ordinary.

Of course, much of the research into the use of real-world metaphor and colour falls to the psychologists [Csinger, 1992], but if computer scientists can arrange to present data in familiar ways, such as movement under the same physical laws as those to which we are accustomed in our world, the brain will unconsciously track much of the change on the display, leaving the conscious mind free to concentrate on the meaning of the data itself.

2.2.2 Data compression

Visualisation techniques allow large amounts of data to be displayed in a very compact form. If each pixel in a 500 × 500 display represents one item, and the colour of the pixel is related to the status of that item, a user can immediately see the status of 250,000 individual items – something which would be impossible using a purely textual display method.

The amount of data generated by today's measurement devices, not to mention the data accessible over the Internet, is growing at an ever-increasing rate. As Barbara Mihalas of the Illinois Supercomputer Centre [Waldrop, 1990] says, 'doing research with this kind of data flow is ... like drinking from the proverbial fire hose'. Visualisation is the only practical way of working with the vast amounts of data available to today's computer users.

2.2.3 Understanding and 'topsight'

Presented with a list of 10,000 figures, it is unlikely that even an expert will be able to discover anything of great significant about the structure of the data. However, if this information is visualised as a line graph, patterns and structures in the data become instantly visible.

The third reason for visualising data is to attempt to gain a deeper understanding of the data itself. By looking at a graphical representation of the data, maybe by using animation to move 'through' the data, a user can appreciate more about the structure and significance of the data than by merely looking at the raw figures.

This is closely related to the concept of *data mining*:

> *Data mining is the search for relationships and global patterns that exist in large databases, but are 'hidden' among the vast amounts of data*
>
> [Holsheimer & Siebes, 1994]

> *Data mining is a term which has become popular to describe a number of techniques for the exploration and exploitation of data. In particular, a large part of data mining involves the visualisation of data and subsequent utilisation of machine-learning techniques for data classification.* [Tattersall & Limb, 1994]

Just as looking at a simple line graph can reveal trends which are not apparent by examining the figures used to generate the graph, so the use of advanced visualisation

tools can bring out aspects of the data being visualised. A user of a visualisation system for monitoring telephone calls across the UK might be able to see patterns of calls which would give an insight into the behaviour of the people connected to the telephone network, and perhaps allow the development of a model of this behaviour.

One step up from insight is what David Gelernter [Gelernter, 1991] calls *topsight* – 'an understanding of the big picture'. If a database can be displayed in some suitable form, either by showing all the data therein or by reducing the quantity of data to a manageable amount of summary information, the user is presented with a chance to appreciate the entire structure. The system then offers the possibility of seeing the global effect of local changes in the data, or of looking for individual data items which share (or don't share) common features.

2.3 Basic Visualisation Techniques

2.3.1 Introduction

This section contains a representative sample of the graph styles offered by *DeltaGraph Pro 3.5*. This is a professional graphing program which is widely used in business to visualise tables of figures.

In the classification of visualisation techniques given below, each method is defined by the number of variables which are shown.

2.3.1.1 Continuous variables

A continuous variable can take any real value between certain limits. For example, the percentage of expenditure spent on a certain product.

2.3.1.2 Discrete variables

A discrete variable can only take a limited range of values, which may or may not be ordered. For example, the colour of a product, or the number of packages it is shipped in.

2.3.1.3 Indices

An index is a discrete variable which divides the database, and for which one data record (usually) exists for each value it takes. For example, the year for which certain figures are given.

2.3.2 One continuous variable, one index

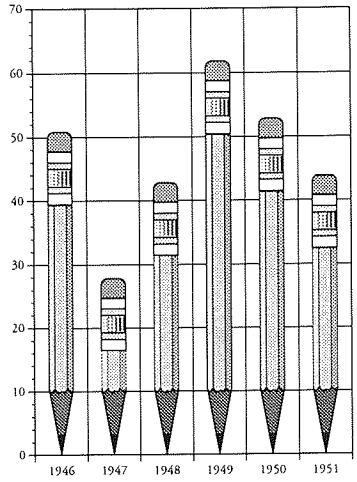


Figure 2.1 – Simple bar chart

A simple bar chart plots the value of one continuous variable against one index variable. The example in figure 2.1 uses pencils to indicate the height of each bar. Tufte [Tufte, 1983] would complain bitterly about the number of lines being used to show just six numbers.

2.3.3 One continuous variable, two indices

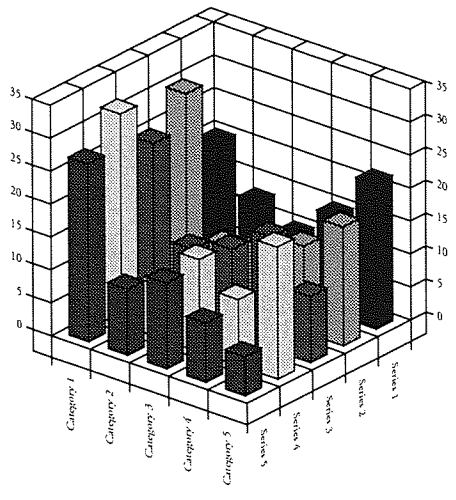


Figure 2.2 – 3-D bar chart

The 3-D bar chart again plots just one continuous variable, but this time using two index variables. The example in figure 2.2 also uses shading to emphasise the values of one of the index variables. Note that smaller bars at the back of the graph can be obscured by larger bars nearer the observer. Once more, a large number of lines is being used to convey a mere twenty-five figures.

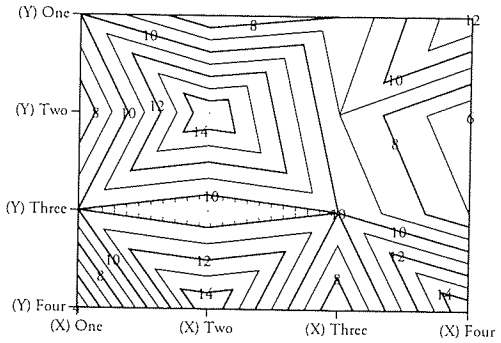


Figure 2.3 – Contour plot

The contour plot shown in figure 2.3 shows contour lines generated from sixteen values located on a discrete grid of index points. Thus it plots one continuous variable against two index variables.

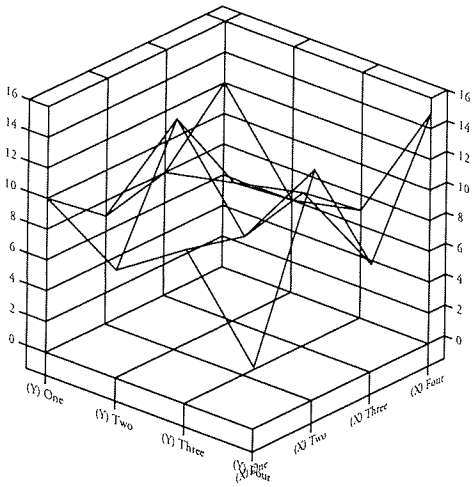


Figure 2.4 – 3-D wireframe surface plot

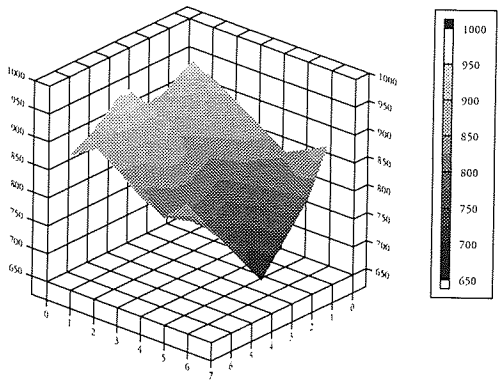


Figure 2.5 – 3-D shaded surface plot

The two 3-D surface plots shown in figures 2.4 and 2.5 also show one continuous variable against two indices, displaying 3-D surfaces interpolated between heights on a fixed 2-D grid.

2.3.4 Two continuous, one discrete variables

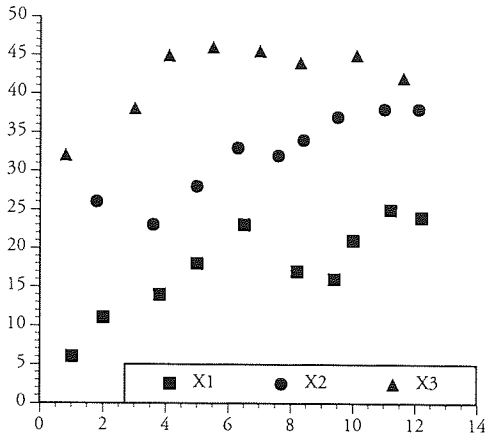


Figure 2.6 – 2-D scatter plot

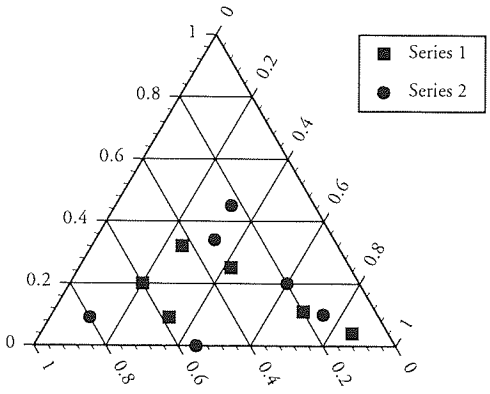


Figure 2.7 – Percentage ternary plot

In both figures 2.6 and 2.7, two continuous variables are used to locate each point plotted on the graph, and a discrete (in this example, binary) variable is used to assign a shape to the point. The ternary plot, though appearing to display three variables, in fact shows only two, due to the constraint that the 'three' variables must sum to 100%.

2.3.5 Three continuous variables

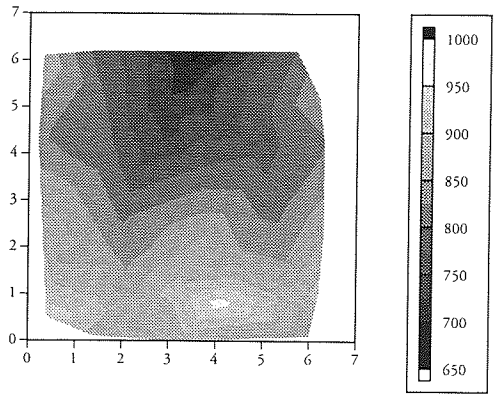


Figure 2.8 – *x*-*y* shaded contour plot

Three continuous variables are used to generate the contour plot shown in figure 2.8. Two locate each point used to generate the contours, and the third gives the 'height' at that point. The values of the first two variables are not explicitly shown – they are similar to index variables, but are not constrained to take values such that every location on the plane is covered.

2.3.6 Three continuous, one discrete variables

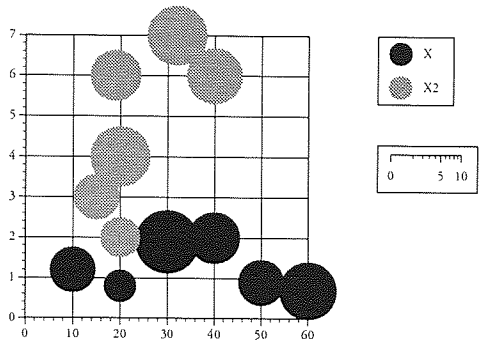


Figure 2.9 – Bubble plot

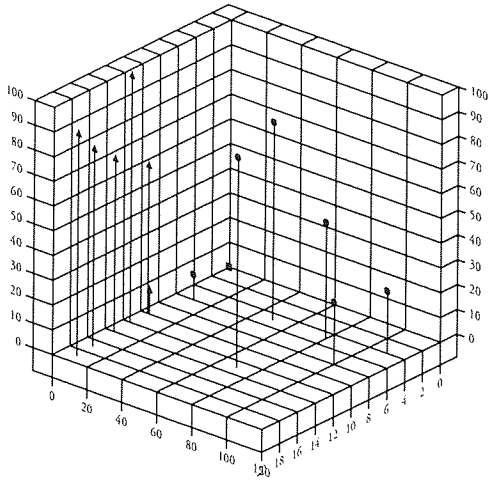


Figure 2.10 – 3-D scatter plot

The bubble plot in figure 2.9 shows three continuous variables (two locate the centre of each bubble, the third defines the bubble radius) and one discrete variable (the colour of the bubble). This is basically a 2-D scatter plot with two additional features shown at each point. The 3-D scatter plot also shows three continuous variables (the location and height of each point) and one discrete variable (the symbol shown at the top of each line).

2.3.7 Three continuous variables, two indices

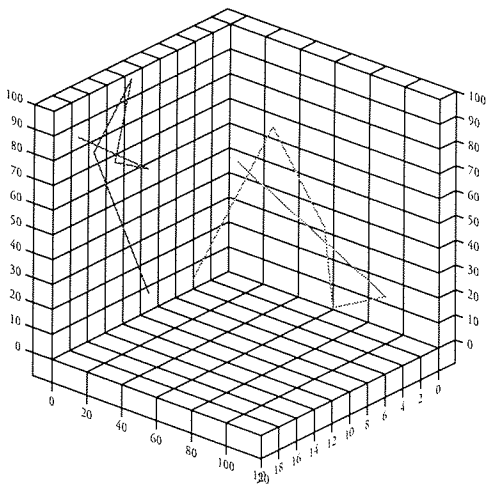


Figure 2.11 – 3-D scatter line plot

An extension of the 3-D scatter plot is shown in figure 2.11, In this plot, three continuous variables are given for each value pair of two index variables, which define a set of lines, and a set of points along the lines. Strictly speaking, neither index

variable is an index in the sense used elsewhere in this section, since the line selection index is unordered, the line position index has ambiguity in its direction, and there is no constraint that the lines have to have an equal number of points defined.

2.3.8 Four continuous variables, one index

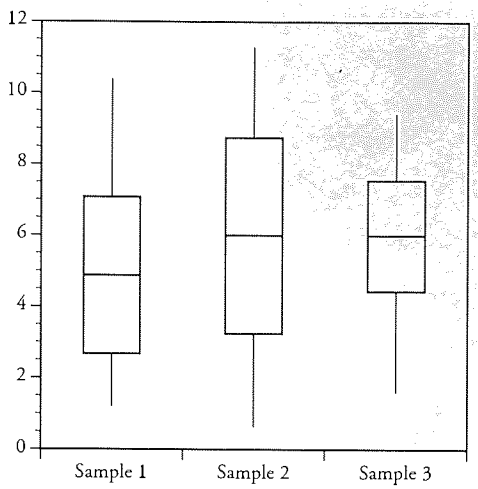


Figure 2.12 – Box plot

The box plot is commonly used in statistical analysis to show the mean, standard deviation, minimum and maximum of a set of variables. Thus a total of four continuous variables are plotted against one index variable. An example is shown in figure 2.12.

2.3.9 Four continuous, one discrete variables

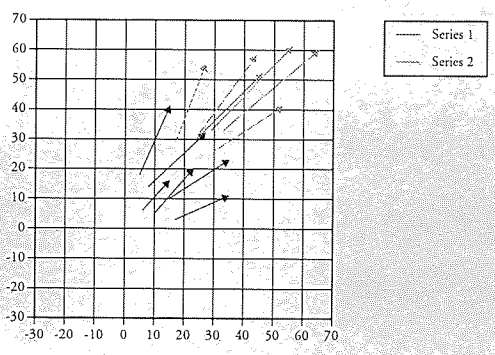


Figure 2.13 – 2-D vector plot

The vector plot shown in figure 2.13 is another way of extending a scatter plot, by showing two continuous variables (either direction and length, or x and y offsets) and a discrete variable (the shade of the arrow), in addition to the two continuous variables which locate that start of the arrow.

2.3.10 *n* continuous variables, one index

The nine graphs shown below in figures 2.14 to 2.22 are all able to show any (reasonable) number of continuous variables against one index. Each has its particular application, whether following trends, analysing changes or comparing variables, and though some enable a reader to easily extract numeric values from the data, some do not.

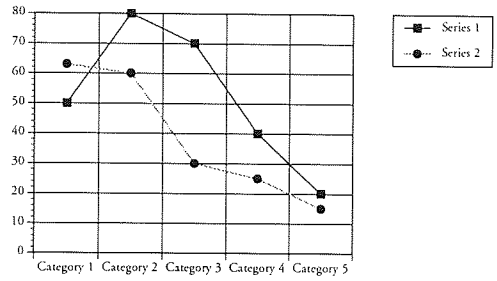


Figure 2.14 – Line graph

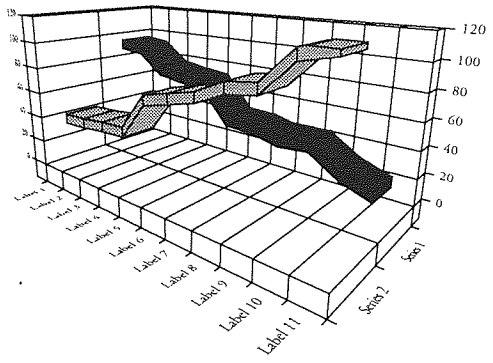


Figure 2.15 – 3-D ribbon plot

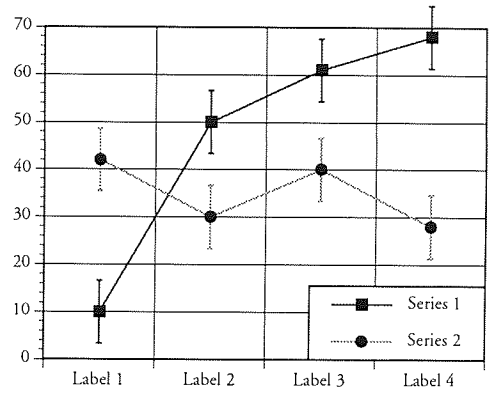


Figure 2.16 – Line graph with error bars

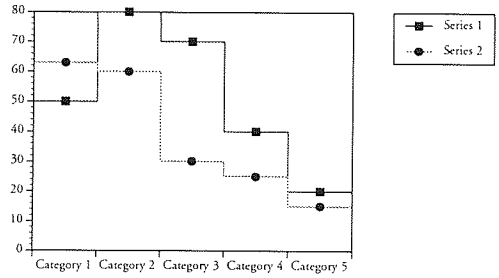


Figure 2.17 – Stepped line graph

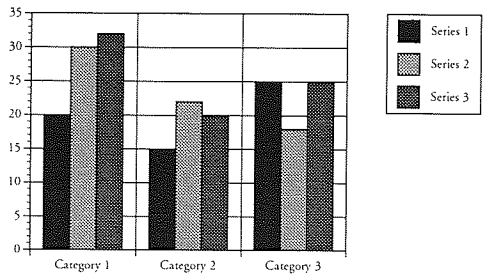


Figure 2.18 – Bar chart

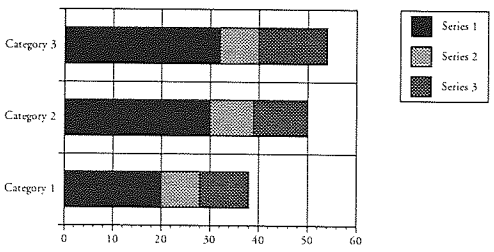


Figure 2.19 – Stacked bar chart

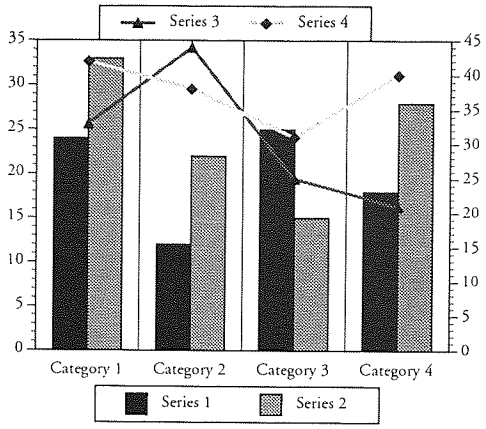


Figure 2.20 – Bar/line overlay chart

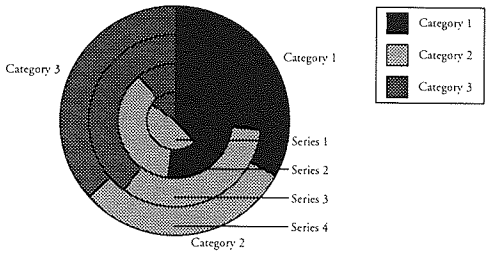


Figure 2.21 – Pie chart stack

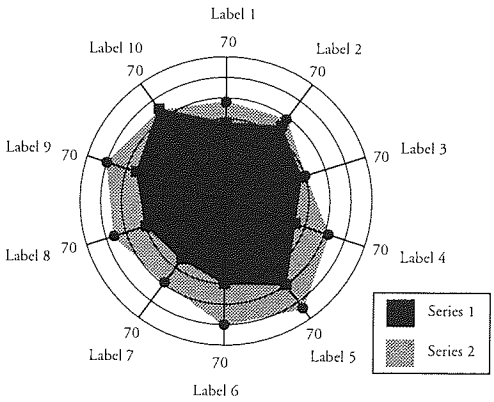


Figure 2.22 – Spider plot

2.4 Advanced Visualisation Techniques

2.4.1 Introduction

> *If progress is to be made in graphics, we must be prepared to set aside old procedures when better ones are developed, just as is done in other areas of science.*
> *[Cleveland & McGill, 1984]*

This section details some of the more advanced visualisation methods which have been developed for examining multi-dimensional data, then looks at some specific visualisation applications and the techniques they use.

2.4.2 General-purpose techniques

2.4.2.1 Andrews curves

An Andrews curve [Andrews, 1972] uses the components of a data vector to draw a curve, using the formula shown below in equation 2.1, where t varies from $-\pi$ to π. The components must evidently be standardised to be of the same magnitude, and often they are ordered by decreasing importance.

$$f_x(t) = \frac{1}{\sqrt{2}}x_1 + x_2 \sin t + x_3 \cos t + x_4 \sin 2t + x_5 \cos 2t + \ldots \qquad 2.1$$

The curves from an entire dataset can then be plotted on one graph, known as an Andrews plot, as shown in figure 2.23, which shows 150 curves generated from the 'Iris' database (a standard four-dimensional database in statistics and neural computing, containing measurements from iris flowers). The graph demonstrates the main use of Andrews plots: to visually identify clustering in the data. Two clusters are clearly visible, suggesting that the database contains two well-separated groups of records. The Andrews plot has the added benefit that Euclidean distances between data points are preserved in the distances between their curves in the plot.

However, it can be seen that Andrews plots are not suited to the visualisation of large databases: even with this small database, the plot is rather dense with information, and takes a very long time to redraw. With ten thousand twenty-dimensional records, the plot would take an age to construct, and would probably be unreadable.

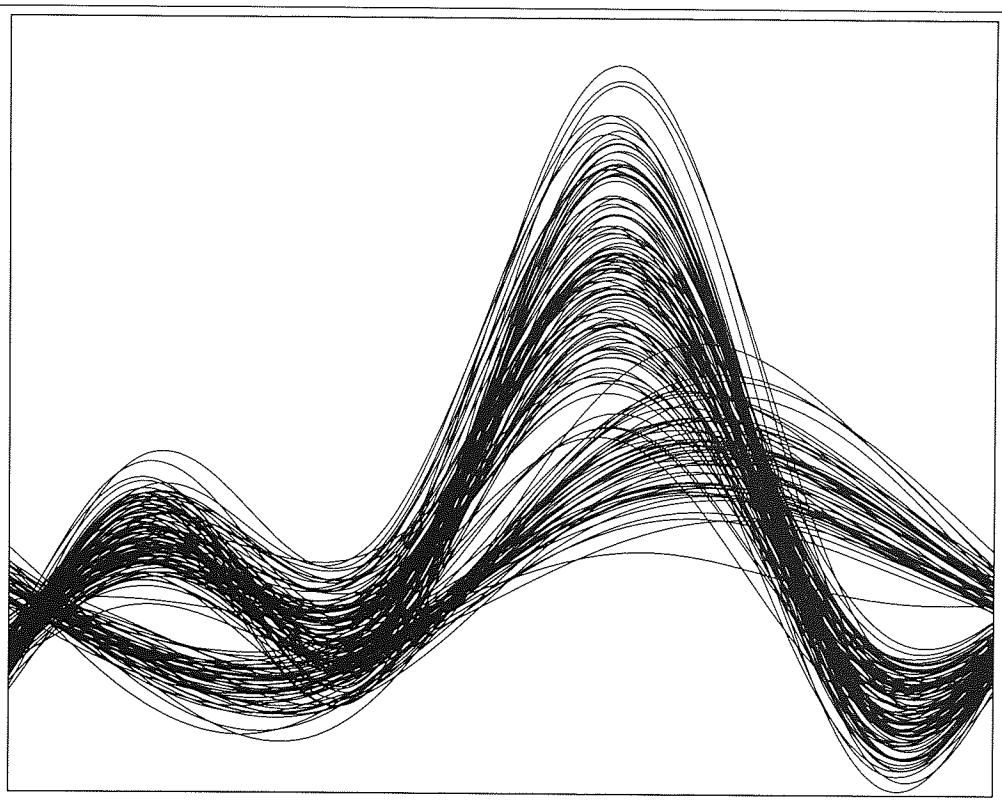


Figure 2.23 – An Andrews plot, showing 150 Andrews curves

2.4.2.2 Chernoff faces

An interesting approach to visualisation which has been much discussed but appears to be of little practical use is Chernoff faces [Chernoff, 1973]. A series of humanoid cartoon faces are drawn, one for each data record, with various features of the face modified according to the values of particular fields of the record. For example, one field might control the angle of the eyebrows, another the size of the ears, and so on. If the assignments of fields to features is appropriately made, the faces can – in theory – be made to look happy, angry, surprised, worried etc when showing suitable data.

In practice, however, a number of variations of assignments of fields to features has to be tested before a suitable match is found, and even then the representation gives results which are interpreted subjectively [Everitt, 1978]. Additionally, each face has to be clearly visible, thereby limiting the number of faces which can be displayed on a screen or page, and in order to identify clustering or trends within the data, the user has to continually compare faces across the display.

For these reasons, Chernoff faces are inappropriate tools for visualising any given large, high-dimensional database.

2.4.2.3 Texture

It is possible to use texture to give a visual representation of a changing quantity across a plane, as long as there are two variables which can be described by positions on the plane (e.g. the plane could represent a real planar slice through a physical space in which measurements were taken).

One method [Levkowitz & Pickett, 1990] is similar in principle to Chernoff faces, but uses tiny line-based icons whose shape alters with the values of up to five data dimensions. The combination of many icons across the plane results in a visual texture; unlike Chernoff faces, individual icons are not meant to be examined.

Alternatively, synthesised greyscale or coloured textures may be used in various ways to represent the variation of data across a plane, or multidimensional data in general [Ware & Knight, 1992; Ware & Knight, 1995].

2.4.2.4 Colour

Colour has great potential to convey additional information when used in combination with existing monochromatic visualisation techniques [Herman & Levkowitz, 1992]. Up to three extra dimensions of data can be incorporated, using any of the common 3-D representations of colour (red/green/blue, hue/saturation/value etc), though precise analysis is difficult as the brain is not designed to split colours into three channels. A simpler solution is to use one colour scale, for example a spectrum from red to violet, or a temperature scale running from blue to red. If feasible, dynamic manipulation of colour mappings has been found to be a useful feature [Rheingans, 1992].

2.4.2.5 Sonification

In addition to the visual sense, it should be possible to use the human ear as an input device. Whereas use of visual information is visualisation, use of audio information can be termed sonification. One augmentation of a visualisation system using sonification [Rabenhorst, 1990] assigns three dimensions of a vector gradient to the de-tuning and stereo balance of each note in a three-note musical chord, in such a way that local minima and maxima can be found without looking at the screen – even by users classed as 'tone-deaf'. Needless to say, systems which rely on sound may require additional computer hardware, and are impractical for an office environment.

2.4.2.6 Lenses

The concept of using 'lenses' to assist in visualisation has been widely explored, often by using a 'fisheye lens' to visualise large structure on small screens by enlarging single or multiple regions of the screen at a time [Sarkar & Brown, 1992; Sarkar & Reiss, 1992].

An impressive development of this concept is the 'table lens' [Rao & Card, 1994; Rao & Card, 1995] which 'supports navigating around a large data space easily isolating and investigating interesting features and patterns'. Plate 2.1 below shows an example of the use of a table lens to examine a database of baseball statistics. It shows 23 statistics for 323 players (each line of the image), containing both quantitative and categorical information, sorted by 'position' and then by 'hits'. The authors claim 'we are able to quickly find interesting correlations or patterns that made sense based on a basic understanding of the domain'.

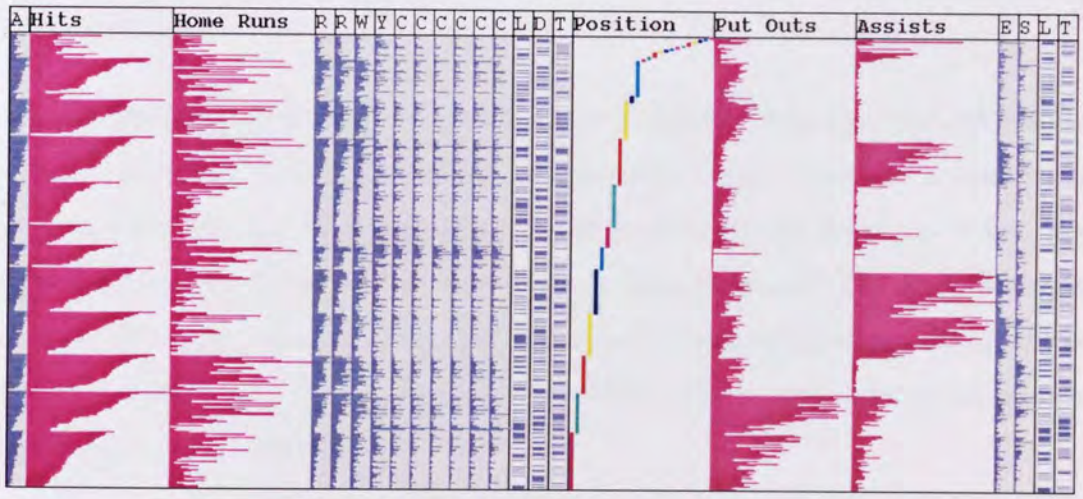


Plate 2.1 – Example of the use of a table lens [Rao & Card, 1995]

Related work on filters has developed a complete family of 'movable filters' [Stone *et al*, 1994] and 'see-through tools' [Bier *et al*, 1994] which can perform a great many tasks on a display, such as adding detail, identifying text properties, changing layering etc. The movable filter has a great deal to offer, not only to data visualisation but to the user interface in general.

2.4.2.7 Three-dimensional methods

A vast amount of interested has been published in the field of three-dimensional visualisation – in the scientific community [e.g. Conner *et al*, 1992; Lee, 1993; Neesham, 1993], in popular books [e.g. Ellis *et al*, 1991; Rheingold, 1991; Sherman & Judkins, 1992] and in science fiction [e.g. Gibson, 1984; Stephenson, 1992; Crichton, 1993] – and the topic is far too wide for a full summary here. Edwards [Edwards, 1992] presents a summary of the processes of visualisation, and describes the 'new generation of interactive data visualisation tools'.

Many techniques, for example *n*-Vision [Feiner & Beshers, 1990] require specialised graphics hardware such as virtual reality headsets, stereoscopic displays, advanced input devices etc. Others use standard display equipment; an overview of some interesting 3-D applications for database visualisation which have been developed in the UK are discussed by Benford [Benford *et al*, 1994].

2.4.2.8 Bertin's ideas

Jacques Bertin's 1977 work *La Graphique et la Traitment Graphique de l'Information* [Bertin, 1981 (translation)], although almost impenetrable, contains a number of quite remarkable ideas concerning data visualisation (although the phrase is not used – at the time, even the spreadsheet had not been invented). He worked almost entirely with huge boxes of 'dominoes' which were reordered ('permuted') by row or column using large sticks; he and his workers would often spend three to six months analysing just one table of data.

Most of the text, though extremely interesting, is far beyond the scope of this thesis, and has little to directly offer to database visualisation using computer graphics. However Bertin did, perhaps, foresee the rise of data visualisation:

> *Graphics is a means of communicating with others. That is its best known application. It can also serve to define and resolve problems of information-processing. This application is now going beyond the realm of specialists and becoming widely available, due to a reduction in technical requirements and to semiological simplification. Moreover, graphics is progressing even farther by giving a visible form to research and methodology.*
>
> *But ... it is necessary to relearn how to 'see'. That is perhaps the essential property of Graphics.* [Bertin, 1981]

2.4.3 Specific visualisation systems

2.4.3.1 IBM Parallel Visual Explorer

IBM's Parallel Visual Explorer (PVE) [Chatterjee, 1995] is a relatively new product which 'enables the unambiguous visualisation of datasets with *many* (in principle unlimited) variables'. It uses a method known as 'parallel coordinates' which maps coordinates as parallel axes (rather than the traditional perpendicular axes), and connects them with a coloured line.

IBM claims that PVE is applicable in a wide variety of multivariate problems, including pharmaceutical research, aerospace, insurance underwriting and oil exploration. Plate 2.2 shows PVE in action, analysing fluctuations in the money markets.

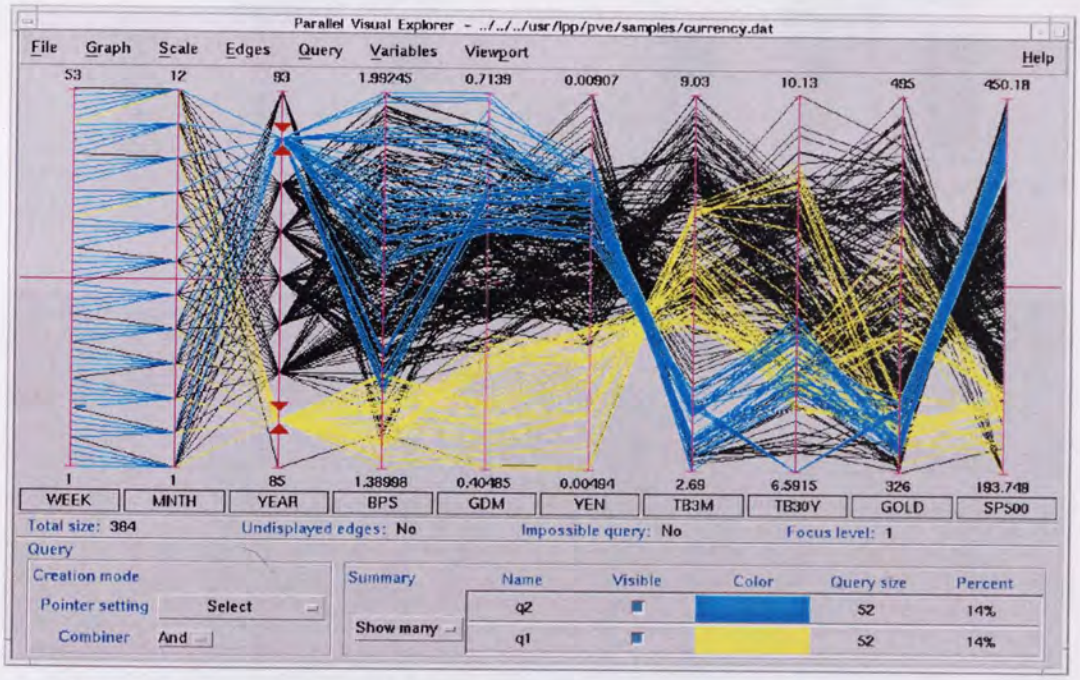


Plate 2.2 – PVE being used to explore the money markets [Chatterjee, 1995]

In the hands of an expert, PVE is demonstrably a powerful tool for analysing high-dimensional databases. However, judging by the example above, it is not immediately clear what the display represents, and it might be difficult to introduce to a non-specialist.

2.4.3.2 GIFIC

GIFIC [Lesser, 1995] is a commercial product which claims to be able to display close to two thousand data elements on a single display, using simple algorithms (which were not disclosed). Plates 2.3 and 2.4 show examples of GIFIC displays. Each character

space is a KEGS (knowledge enhanced graphical symbol), coloured according to one of nine levels from 'panic low' (yellow) through 'normal' (green) to 'panic high' (red). Plate 2.3 shows measurements from two human hearts: evidently the one on the right is abnormal. Plate 2.4 shows one hundred combat troops; it appears that two are in difficulty and one may be dead.

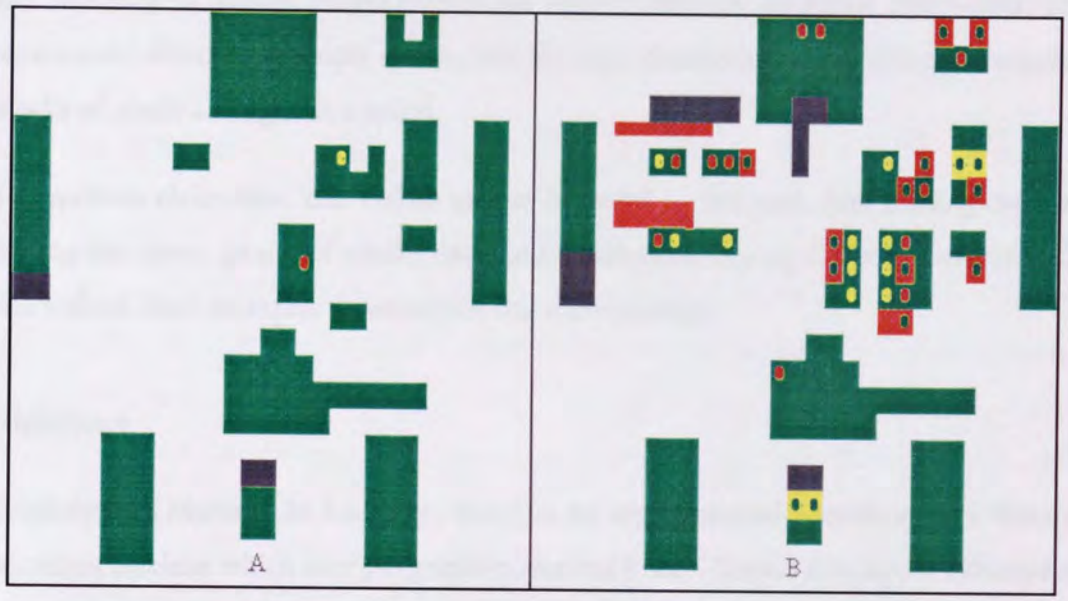


Plate 2.3 – GIFIC display showing two hearts [Lesser, 1995]

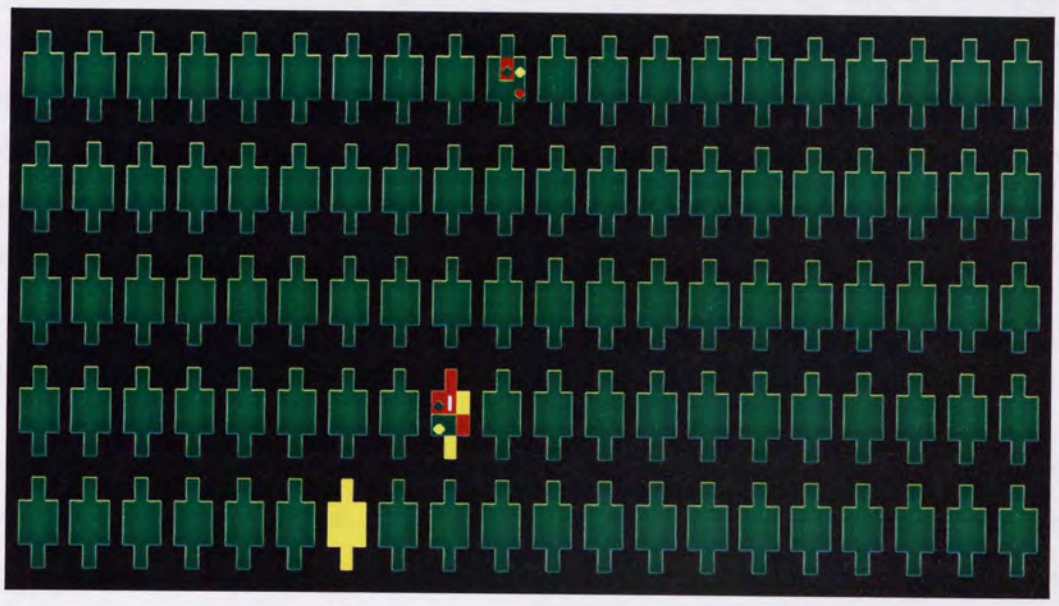


Plate 2.4 – GIFIC display showing 100 combat troops [Lesser, 1995]

It is evident that GIFIC requires considerable knowledge of the problem domain which a new database models before it can generate plots – for example, which measurements (or combinations of measurements) constitute a 'normal' heart? Given a new database from a different domain, GIFIC is unlikely to be able to immediately assist in visualisation.

2.4.3.3 VisDB

VisDB [Keim & Kriegel, 1994] is a visualisation system developed for presenting the results of queries of high-dimensional relational databases. The pixels on the display each represent a data item. They are arranged in rectangular spirals, ordered by the relevance to the query (more relevant items are located at the centre of the spirals), and coloured to denote distance from the correct answers. Multiple dimensions are represented either by multiple spirals, one for each dimension, or by using rectangular blocks of pixels arranged in a spiral.

The authors claim that 'the VisDB system is useful ... for such data mining tasks as finding hot spots, groups of similar data, and correlations among different dimensions', and indeed their examples demonstrate this convincingly.

2.4.3.4 TripleSpace

TripleSpace [Mariani & Lougher, 1992] is an experimental interface to a binary relational database which uses 3-D graphics, derived from Gibson's concept of cyberspace (see chapter 3). Each item in the database is shown as a cube, its faces labelled with the three text of the relationship it represents. The user can 'fly' around the space, viewing the cubes from all angles and visually examining both individual data items and the relationships between groups of items. Queries can be made, resulting in planes, vectors or points in the space, though as the authors point out, 'this no longer really makes full use of the 3-D, although we can still 'fly' around the information' – a similar result will be seen later in chapter 3.

2.4.3.5 Telecommunications visualisation

Engineers working for BT have investigated the applications of visualisation to telecommunications network data [Walker *et al*, 1993]. They use techniques including overlays on maps of the UK, coloured 3-D bar charts and multiple deformed sheets. Animation is also applied, for example to investigate the effect of a lightning strike on the network.

However, the paper describes only what might be termed 'standard' scientific visualisation techniques, and introduces nothing particularly novel.

2.4.4 The Xerox Information Visualiser

Xerox PARC has long been a centre of exciting work on the human-computer interface. One result has been the development of the Information Visualiser [Card *et al*, 1991; Clarkson, 1991; Robertson *et al*, 1991B]. This system which uses a variety of 2-D and 3-D techniques to present information on a computer screen, including real-time animation and three-dimensional depth cueing.

2.4.4.1 The perspective wall

One visualisation technique is the perspective wall [Mackinlay *et al*, 1991]. This shows data such as a time-line, which traditionally is difficult to visualise without either showing all the time-line and not being able to see any detail, or concentrating on one section and not displaying the rest. The perspective wall overcomes this by showing one section enlarged in the centre of the screen, with the rest of the wall receding into the distance at either side.

Plate 2.5 shows a perspective wall in use as part of Xerox's *Visual Recall* product.

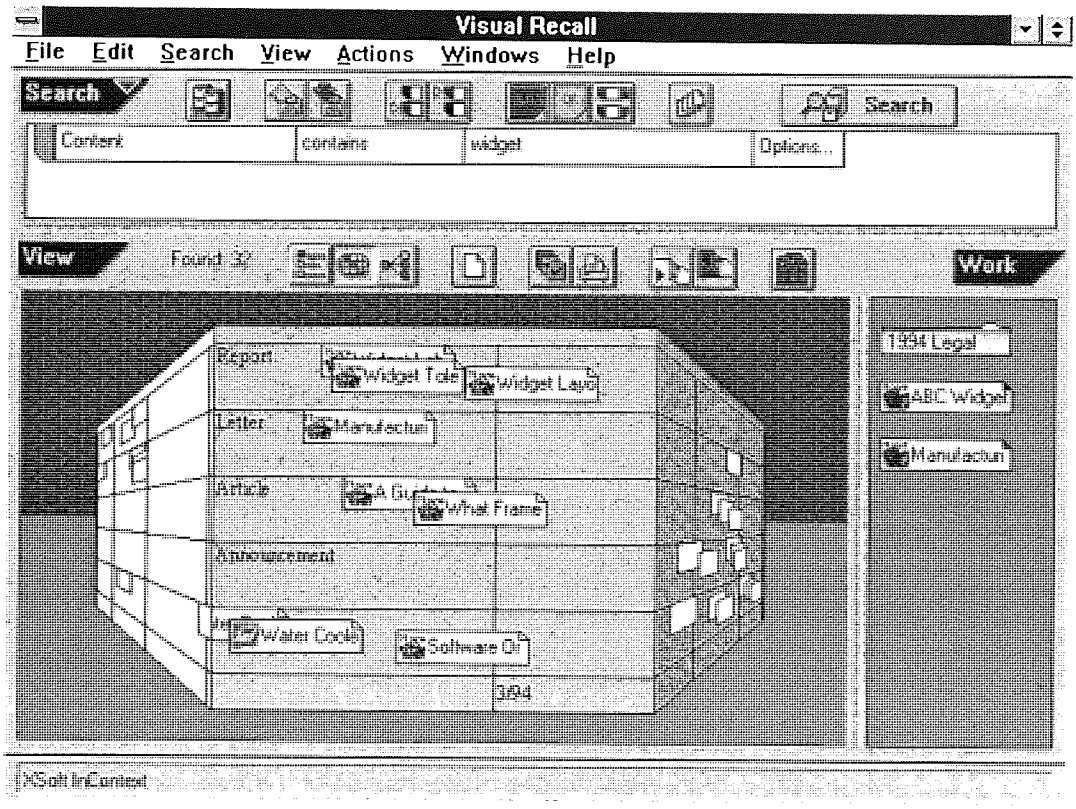


Plate 2.5 – Visual Recall grid view, based on the perspective wall

2.4.4.2 Cone and cam trees

The other important new visualisation technique introduced in the Information Visualiser is concerned with hierarchical information such as organisation charts. The cone tree [Robertson *et al*, 1991A] represents the hierarchy using 3-D graphics, where each node's subordinate nodes are arranged in a horizontal circle beneath it. The cam tree is similar, but horizontally aligned, with each node's subordinates arranged in a vertical circle to its right.

The cone/cam tree allows a great deal of information to be shown on a single display. To examine parts of the tree, the user can click on a node, and the entire tree rotates in real-time to bring the node and its path to the 'top' of the tree to the front of the display. Sections of the tree can be 'pruned' away for clarity, and 'grown' back again if required.

Plate 2.6 shows a greatly-simplified version of a cam tree in use in *Visual Recall*.

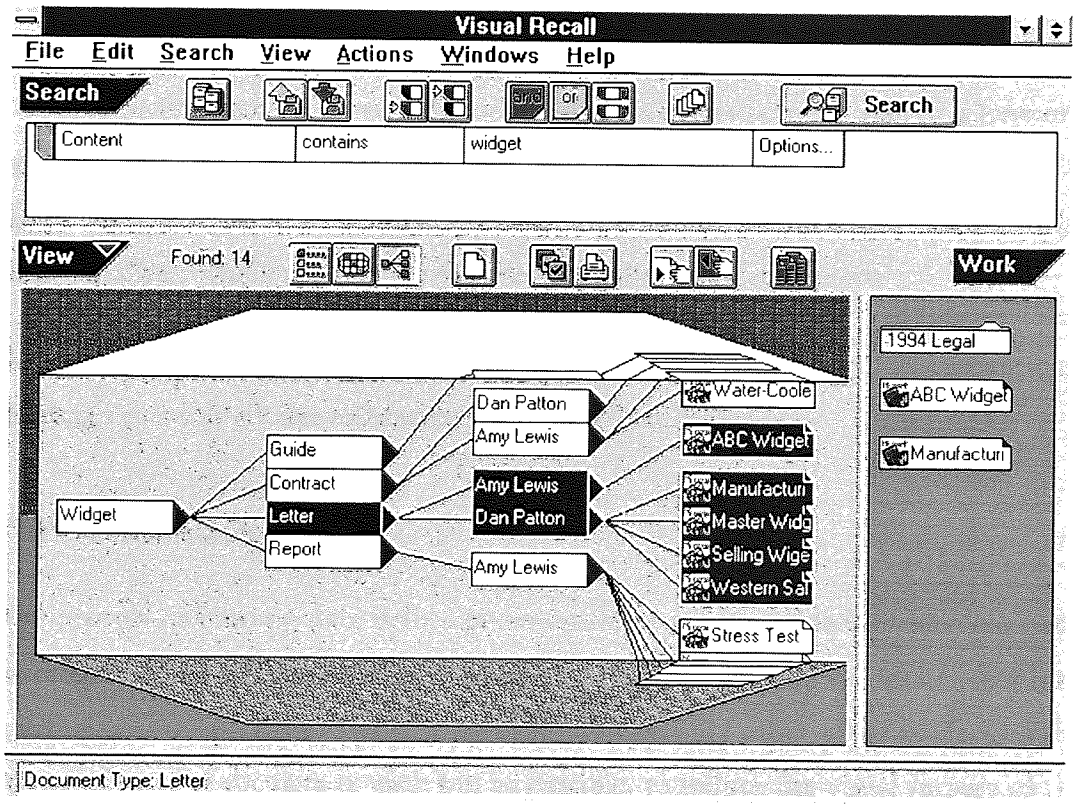


Plate 2.6 – Visual Recall tree view, similar to a cam tree

Cone trees have been proved to be a very popular technique and have been widely used outside Xerox [Conner *et al*, 1992; Tversky *et al*, 1993].

2.4.4.3 Architecture

Needless to say, the 3-D real-time animation required by the Information Visualiser demands a dedicated computer architecture which allows the system to continue to feel 'alive' even while completing a complex processing task. This architecture is known as the 'cognitive coprocessor architecture' [Robertson *et al*, 1989], and runs on a Silicon Graphics Iris (a powerful workstation with dedicated graphics hardware).

2.4.4.4 Conclusions

Though extremely interesting, the visualisation techniques of the Information Visualiser offer little to the area in question: the perspective wall analyses time-line-based data, and the cone/cam tree relies upon a hierarchical database – neither can directly be applied to the exploration of large tabular databases.

2.4.5 Scatter plot matrix systems

Chapter four of this thesis will introduce 'MADEN,' a visualisation system derived from a matrix of scatter plots. The idea of such matrices is by no means a new one [e.g. Chambers *et al*, 1983], and has been used in many ways [e.g. Becker *et al*, 1988; Scott, 1991], though most of these systems were evidently designed for use only with small databases where it is feasible to generate traditional scatter plots.

Some visualisation systems which use a matrix of plots to display large multi-dimensional databases in a way similar to MADEN are summarised below.

2.4.5.1 Carr's binned data plots

The process of binning data before presentation in a form of density plot has been investigated [Carr, 1991]. Carr uses hexagonal bins, in order to give clearer plots than rectangular bins produce. His varying-sized hexagons are offset to lie as close to the centre of mass of the data in each bin as possible, to reduce the visual impact of the lattice and better show the accurate data density.

This technique is more suited to print media than display screens, as it is monochromatic and requires high resolution to show the hexagons at small sizes. On a computer screen, a greyscale plot (as used in MADEN) is likely to be preferable.

2.4.5.2 Mihalisin's method

A novel technique for visualising multidimensional data [Mihalisin *et al*, 1991] uses a matrix of plots which show the relationship between two or more variables in each plot through binning and averaging. From the examples in the paper, it seems that the correlations between variables can be clearly seen, but the fine detail of the MADEN density plots is absent.

2.4.5.3 Boyle's n-dimensional visualisation system

A new paper reviewing 3-D visualisation systems [Boyle, 1995] concludes that in many cases a 2-D representation is superior (a similar conclusion will be presented at the start of chapter four). A 2-D matrix of scatterplots, apparently similar to MADEN's, is presented, as shown in plate 2.7. No further details regarding the techniques used were available.

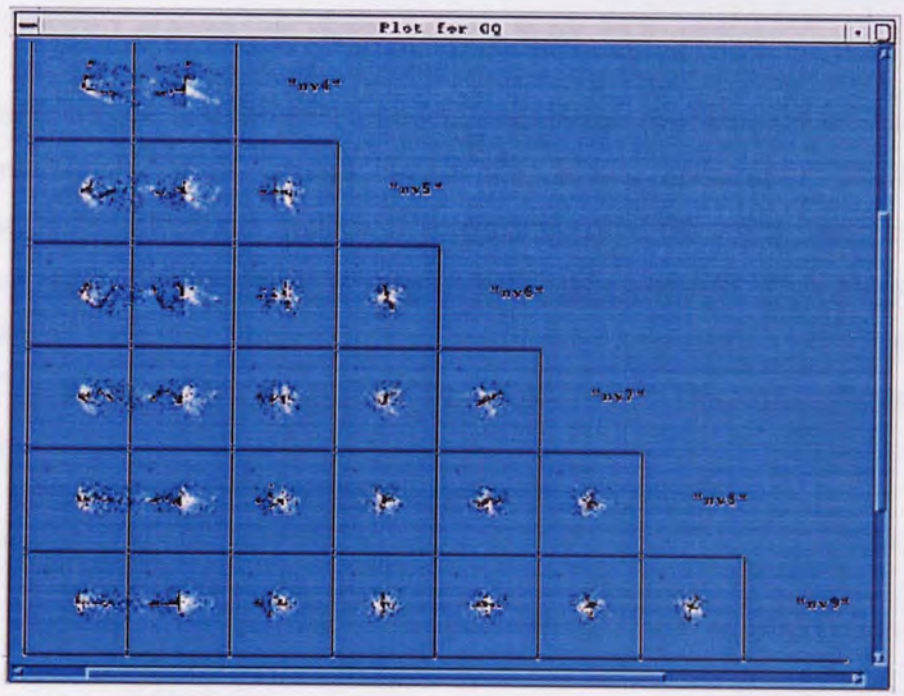


Plate 2.7 – Boyle's *n*-dimensional visualisation system [Boyle, 1995]

2.4.5.4 Tweedie's prosection matrix

The 'prosection view' [Furnas & Buja, 1994; Tweedie, 1995] is 'a composition of a section and a projection applied to a high-dimensional database, in order to reveal internal structure of intermediate dimensionality'. The diagram in plate 2.8 overleaf shows the process of prosection, with a section (along p3) of the three-dimensional

space being projected onto two axes (p1 and p2). It will be seen to be equivalent to the 'dependent walls' of the Benediktine cell (chapter three) and the 'dependent enlargements' of MADEN (chapter four).

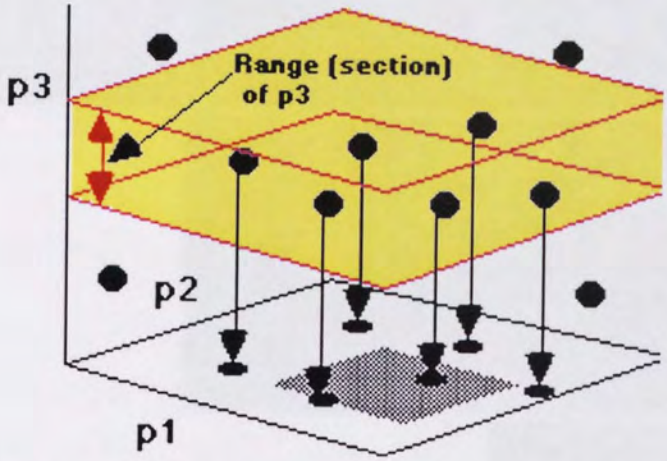


Plate 2.8 – Tweedie's explanation of prosection [Tweedie *et al*, 1996]

A paper to be published next year [Tweedie *et al*, 1996] describes the 'prosection matrix': a matrix of prosection views, as shown in plate 2.9 overleaf.

The prosection matrix is designed for examining abstract mathematical models, rather than databases of real information, but has considerable similarities to MADEN. The red areas in the matrix are 'regions of acceptability' and the yellow selection is the 'tolerance region', which can be adjusted visually and numerically to fit the region of acceptability. Each view in the matrix shows the projection onto the two axes of the view of the section of the data defined by the yellow selection rectangles on all the other axes.

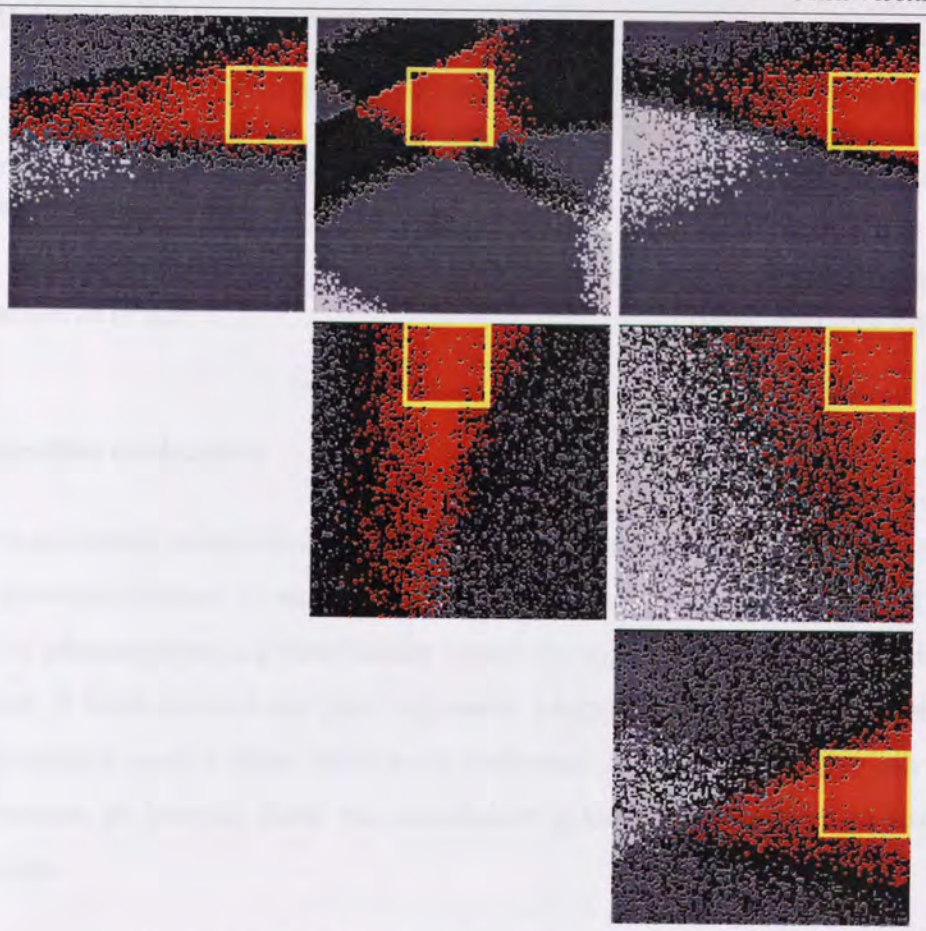
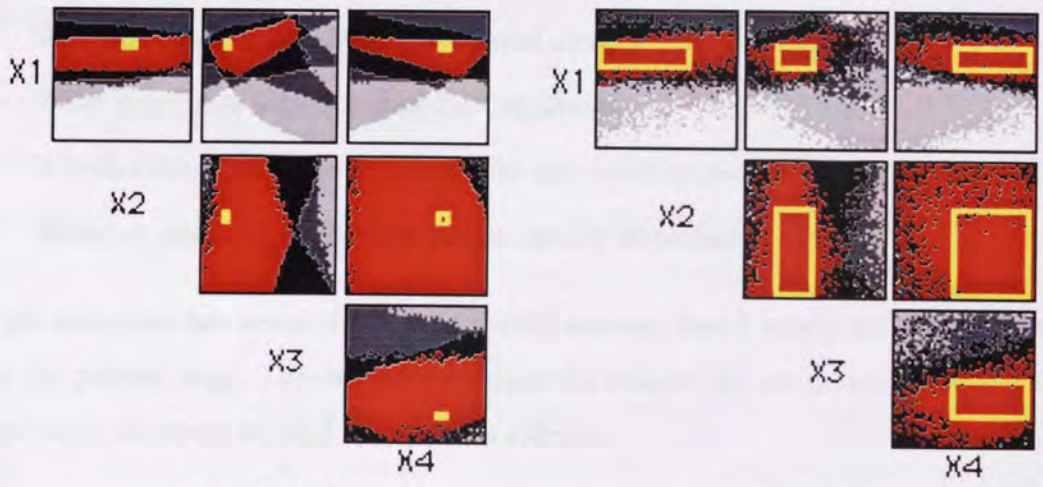


Plate 2.9 – The prosection matrix [Tweedie, 1995]

In an obvious parallel of selecting customers from the mail or finance databases in MADEN and examining their average response, the prosection matrix system measures the 'yield' – the percentage of the tolerance region which is acceptable. The yield of the tolerance region in plate 2.9 is 92%; plate 2.10*a* shows a yield of 100%; plate 2.10*b* has a wider tolerance yet retains a yield of 98%.



a – 100% yield *b* – 98% yield

Plate 2.10 – Adjusting the tolerance on the prosection matrix [Tweedie *et al*, 1996]

2.5 Conclusions

> *[Data visualisation] provides ... two considerable advantages. First the data can be accessed at any level – as opposed to ... answer[ing] specific enquiries. ...Second, the data can be displayed as graphics, or maps, as opposed to endless rows of numbers.*
>
> *...By displaying information in a graphical mode it is possible to help the brain function at its best.* [Ballé & Jones, 1993]

2.5.1 Comparative evaluation

Figure 2.24 overleaf assesses six of the more promising visualisation techniques described in the previous sections, by testing whether they provide each of nine features which would be advantageous in a visualisation system for exploring large high-dimensional databases. A filled circle in the chart represents a supplied feature; a half-filled circle an intermediate result – either because the technique in question only partially fulfils the criterion, or because there was insufficient information available to make an evaluation.

The nine criteria are:

- Performance with a database of around six dimensions (k)
- Performance with a higher-dimensional database, with twenty or so dimensions
- Performance with a database with a few hundred records (n)
- Performance with a large database with ten thousand or more records
- Whether discrete data fields are handled well
- Whether quantitative results can be read directly from the display
- Whether records with identical data are shown
- Whether correlations between variables can easily be picked out
- Whether clustering in the data can be visually identified

Each technique was assessed in a purely static manner, based purely on its appearance on the printed page. This is necessary since the author had no means of testing the interactive elements of cited visualisation software.

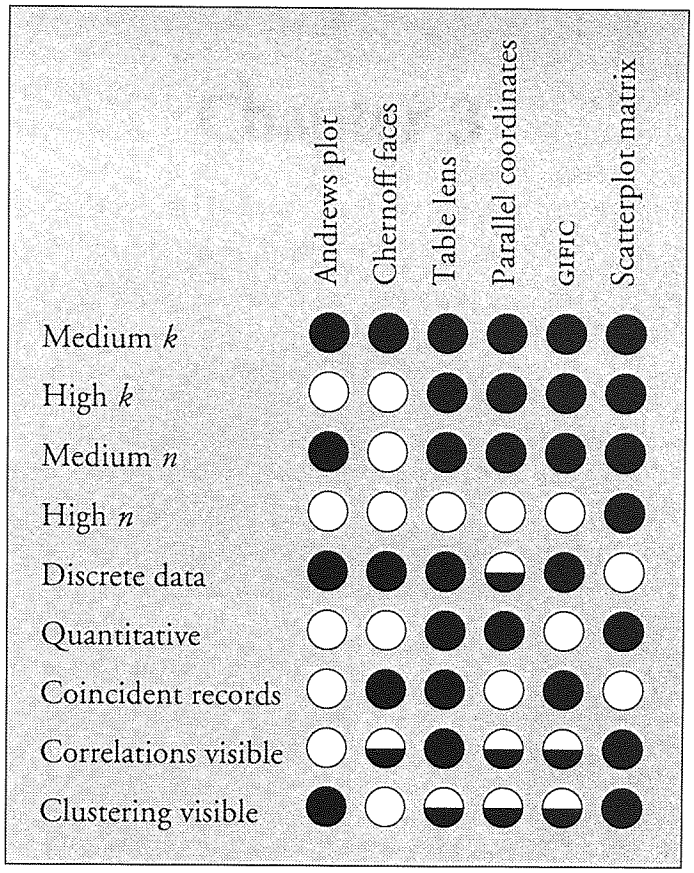


Figure 2.24 – Comparison of features of six visualisation techniques

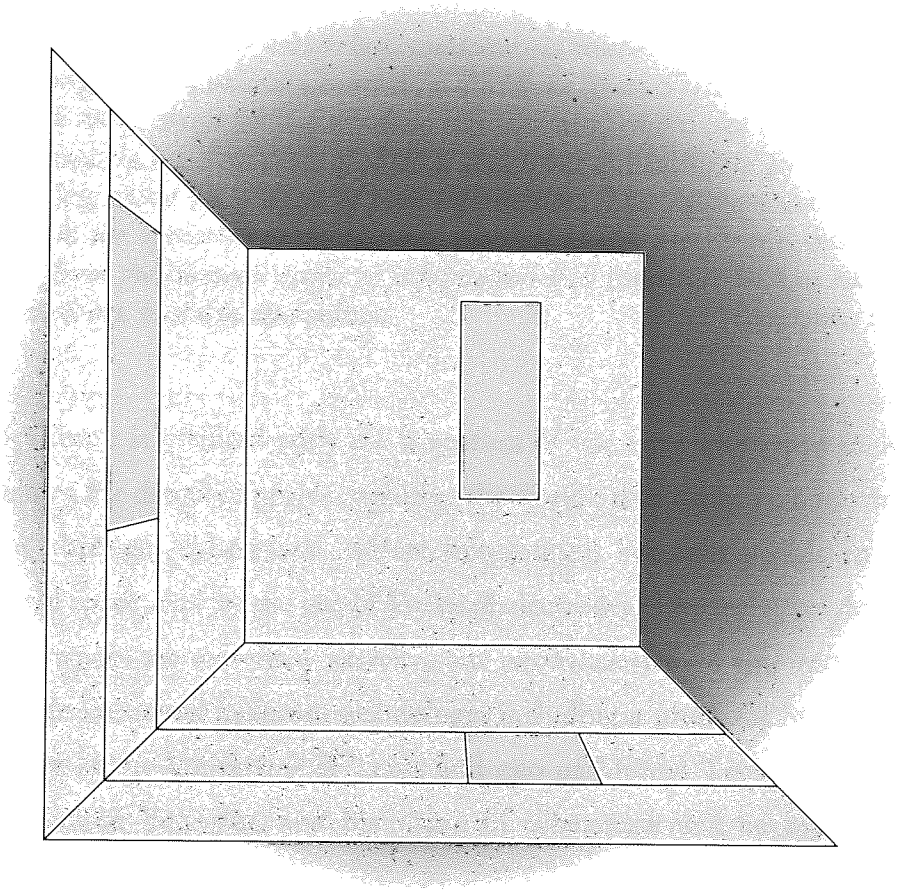
Examination of the table shows that while most techniques are suited to small databases of small dimensionality, few are applicable to large high-dimensional databases. Only the three visualisations where a separate entry is shown for each record are able to distinguish coincident records, and two of these are unable to provide quantitative information.

2.5.2 Summary

It appears clear that there is a need for a new visualisation tool which addresses the shortcomings which have been identified in the existing techniques. Of course, such a tool should additionally provide many more features through interaction with its users, enabling the database under examination to be explored, rather than merely viewed.

The next chapter introduces William Gibson's concept of 'cyberspace', which is a very alluring approach to visualisation, and describes an implementation based upon some of his ideas.

Chapter 3



Visualisation Tools I: The Benediktine Cyberspace Cell

We have first raised a dust and then complain we cannot see. [Berkeley, 1710]

3.1 Introduction: What is Cyberspace?

> *Cyberspace. A consensual hallucination experienced daily by billions of legitimate operators, in every nation, by children being taught mathematical concepts ... A graphic representation of data abstracted from the banks of every computer in the human system. Unthinkable complexity. Lines of light ranged in the nonspace of the mind, clusters and constellations of data. Like city lights, receding...*
>
> [Gibson, 1984]

> *Cyberspace is a globally networked, computer-sustained, computer-accessed, and computer-generated, multidimensional, artificial, or "virtual" reality. In this reality, to which every computer is a window, seen or heard objects are neither physical nor, necessarily, representations of physical objects but are, rather, in form, character and action, made up of data, of pure information. This information derives in part from the operation of the natural, physical world, but for the most part it derives from the immense traffic of information that constitute human enterprise in science, art, business, and culture.* [Benedikt, 1991a]

William Gibson is credited with the invention of the term *cyberspace*, in his book *Neuromancer*. He describes people exploring the world's data using a computer console to 'move' through 'cyberspace', where information becomes visible in a three-dimensional space seen by the use of forehead electrodes. The technical descriptions of the cyberspace are extremely sketchy, but nevertheless fired the imaginations of many people interested in future technology, including a professor in the School of Architecture at the University of Texas and CEO of Mental Technology, Inc. His name is Michael Benedikt, and his vision of cyberspace will be explored in the remainder of this chapter.

3.2 Benedikt's Cyberspace

Michael Benedikt's excellent chapter in the ground-breaking *Cyberspace: First Steps* [Benedikt, 1991B] was in many ways the inspiration for this research. Here, Benedikt details his thoughts on the mechanics of cyberspace, maintaining as many of the physical world's properties as possible while augmenting them to suit a computer-generated, multiple-user cyberspace containing vast amount of data, and introduces seven 'principles' to govern the behaviour of his cyberspace.

Key to Benediktine cyberspace are the concepts of intrinsic and extrinsic dimensions, and the process of unfolding one into the other.

3.2.1 Intrinsic and extrinsic dimensions

Any object located in space can be said to have *intrinsic* properties – its shape, its colour etc – and *extrinsic* properties – its location in space (and time). A pure mathematical point, by definition, has no intrinsic properties, but any real, observable, object must have some intrinsic properties. Location is 'extrinsic' because changing it (by moving the object) has no effect upon the object itself, which retains all its intrinsic properties.

Consider the representation of an n-dimensional dataset in a p-dimensional space ($p < n$; typically $p = 2$ or 3) where each data point is represented by a 'data object' in p-space

The values of exactly p of the n original dimensions can be used as coordinates to locate each data object in p-space. These dimensions are therefore known as *extrinsic dimensions.*

The remaining $n - p$ dimensions are then *intrinsic dimensions*, some or all of which are mapped to intrinsic properties of the data object, such as shape, size, changing size (frequency, amplitude, direction), directional alignment, rotation (direction, speed), colour (up to three dimensions per coloured part of the object), texture etc. Conceptually, any remaining intrinsic dimensions are encapsulated by the data object in a non-observable form.

3.2.2 Unfolding

Once a representation of a dataset has been created in the p-dimensional space as above, a location in p-space can be specified, thereby fixing the values of the p extrinsic dimensions in n-space. As long as there is data at that location in p-space, one or more data points which lie at that location (i.e. which have the given values along these dimensions) are thus 'selected'.

It is then possible to create a new p-dimensional space containing only the selected data points. The extrinsic dimensions of the new space are selected from the previously intrinsic dimensions. Benedikt terms this process *unfolding*, since p intrinsic dimensions have been 'unfolded' from the selected data point(s) and made into extrinsic dimensions in a new space.

Furthermore, as the selection of data points in the original, 'outer', p-space is changed, the data objects in the 'inner' p-space can change, tracking the p newly-extrinsic dimensions of the data objects selected in the outer space.

The unfolding process can evidently be repeated until there are no dimensions left to unfold.

3.2.2.1 Example

Suppose we have a seven-dimensional dataset with dimensions $\mathbf{a}_1 \ldots \mathbf{a}_7$ which we are visualising in a 3-D space with coordinate axes **x**, **y** and **z**, using coloured points. Dimensions $\mathbf{a}_1$, $\mathbf{a}_2$ and $\mathbf{a}_3$ might be assigned to the three extrinsic dimensions, leaving $\mathbf{a}_{4\ldots7}$ as the four intrinsic dimensions, one of which, say $\mathbf{a}_4$, is visible as the colour of the point (on a one-dimensional colour scale).

If a point in the 3-space is selected, this fixes the values along **x**, **y** and **z**, and thereby the values along $\mathbf{a}_1$, $\mathbf{a}_2$ and $\mathbf{a}_3$. Assume that ten data points share these coordinates. We can then unfold three of the intrinsic dimensions, say $\mathbf{a}_4$, $\mathbf{a}_5$ and $\mathbf{a}_6$, as the extrinsic dimensions of a new 3-space, leaving $\mathbf{a}_7$ as the only intrinsic, colour, dimension. This new space contains only the ten data items which were selected in the outer space.

By changing the selection in the outer space and observing the changes in the inner space, it might become clear that movement parallel to $\mathbf{a}_2$ has no effect on the position of data objects in inner space, though their colours change. This would lead to the conclusion that $\mathbf{a}_4$, $\mathbf{a}_5$ and $\mathbf{a}_6$ are independent of $\mathbf{a}_2$, but that there is a relationship between $\mathbf{a}_7$ and $\mathbf{a}_2$.

Thus by utilising one unfolding operation, we have succeeded in visualising all seven of the dimensions of the dataset, and are able to make observations of dependence between the dimensions.

3.2.3 Benedikt's cell

Benedikt then describes his view of a three-dimensional 'room' in cyberspace which can be used to visualise large databases:

> *It is common in scientific visualisation to present three- and four-dimensional data points in a coordinate system space where three of the dimensions are extrinsic, that is, are reflected in/as position in the space, and one is intrinsic (usually colour).*
>
> *Using this method, the space itself is often opaque with its own data, filled with a solid fog. Indeed, it is more like a solid object than a space. It must be peeled and sliced to see within it...*
>
> *In our scheme, this problem is avoided. The walls of the room – we call it a "cell" – correspond to the three planes constituting a conventional, rectangular, and Cartesian coordinate system, namely, the horizontal plane, the "floor" (X-Y), and two vertical planes, the "walls" (X-Z) and (Y-Z). ... The space in the cell itself is almost empty: "almost," because it contains* us, *our* vehicle, *and a* probe...
>
> *Now, the walls of the cell are not blank but display ... the* amount *of information ... that is selected by that particular value of X and Y, or X and Z, or Y and Z as the case may be, and under one of two user-definable conditions: (1)* regardless *of the currently selected value of the remaining third dimension, or (2)* given *the currently selected value of the remaining third dimension.* [Benedikt, 1991B]

How can a display on a 'wall' indicate the amount, or *density*, of information at a particular location in a database? Once again, Benedikt supplies a novel method, outlined below.

- If both axis variables are continuous, the data density can be shown as a coloured image field, for example a greyscale, with black representing no data at those coordinates and white representing the maximum density of data.

- If one axis variable is continuous and the other discrete, a series of ribbons of variable width can display the required information. There is one ribbon for each value of the discrete variable, whose width varies continuously with the other variable.

- If both axis variables are discrete, a plane of discrete rectangles of varying size may be used. A large rectangle would represent a large data density, a missing rectangle (or a point) would represent no data at those coordinates.

In effect, the entire database is *projected* onto the two-dimensional plane defined by the two axis variables, and the resulting density map is used to generate an image on the wall. This is very like a 2-D scatter plot projection, but with density information made explicit (in a standard scatter plot, coincident points appear to be one point).

3.3 Implementation

A visualisation tool was developed to implement the Benediktine cell specifically in order to visualise tabular databases of the kind described in chapter one. This section describes the features of the resultant system.

3.3.1 Platform

As with all the work described in this thesis, this program was written in C++ on Sun SparcStations running the Solaris operating system. The user interface uses the OpenLook *XView* library.

3.3.2 Data input

A general-purpose data file reader was written to enable most databases to be easily read into the program. It reads a header line containing field names, then the data, one record per line. Once the data has been read, each field is automatically assigned a type (continuous, integer or categorical), and its range, maximum value and minimum value are calculated. In the case of categorical variables, the initial letters of each category are used for identification. The letters are sorted into alphabetical order and assigned sequential integer values.

The database is stored in two data structures: a large matrix of floating-point numbers (one row per record and one column per field; integer values are stored as their floating-point equivalents) and an array of structures containing information about each field.

3.3.3 Density projection

The wall displays rely on matrices of density information, generated by projecting the database onto two specified axes. The projection routines accomplish this, generating an output matrix whose contents range from 0 to 1, 0 being no data and 1 being the maximum density value in the particular projection.

The matrix is generated by taking each data record in turn, calculating the matrix indices based on the value of the two specified fields, and incrementing that element of the matrix. Finally, the entire matrix is divided by the maximum value therein.

The size of the output matrix is determined by the type and range of the fields used to generate the projection. An arbitrary maximum size is defined (currently fifty, which gives a usable balance between resolution and speed), since continuous fields evidently cannot be projected onto a continuous density map, as this would require an infinitely large matrix. If a continuous field, or a discrete field with more than this maximum number of values, is used, the axis is 'binned' into the specified number of locations.

3.3.4 Display

A standard 3-D one-point perspective projection algorithm [Watt, 1989; Berger, 1986] was used to generate the display seen by the user of the system, though to simplify calculations the viewing direction was constrained to be parallel to the floor of the cell – i.e. vertical lines in the cell are always projected as vertical lines on the screen. Figure 3.1 overleaf shows the initial view of the mail database.

3.3.4.1 Cell layout and display

The three walls are shown hanging in white space. After experimentation, it seemed the best choice of axis orientation was to use a common origin at the corner of the cell, where all three walls meet. The axes were assigned so that when looking at one wall of the cell (the 'back wall'), with the other wall on the left (the 'side wall'), the back wall displays x in the horizontal direction and y in the vertical direction. The z axis, therefore, points out of the screen towards the user[1]. The initial view was set to be looking directly at the centre of the back wall, at a distance where the side wall and floor are visible.

[1]This differs from Benedikt's assignment of the floor as the x-y plane, but coincides with the 3-D graphics convention of assigning z to the out-of-screen direction.

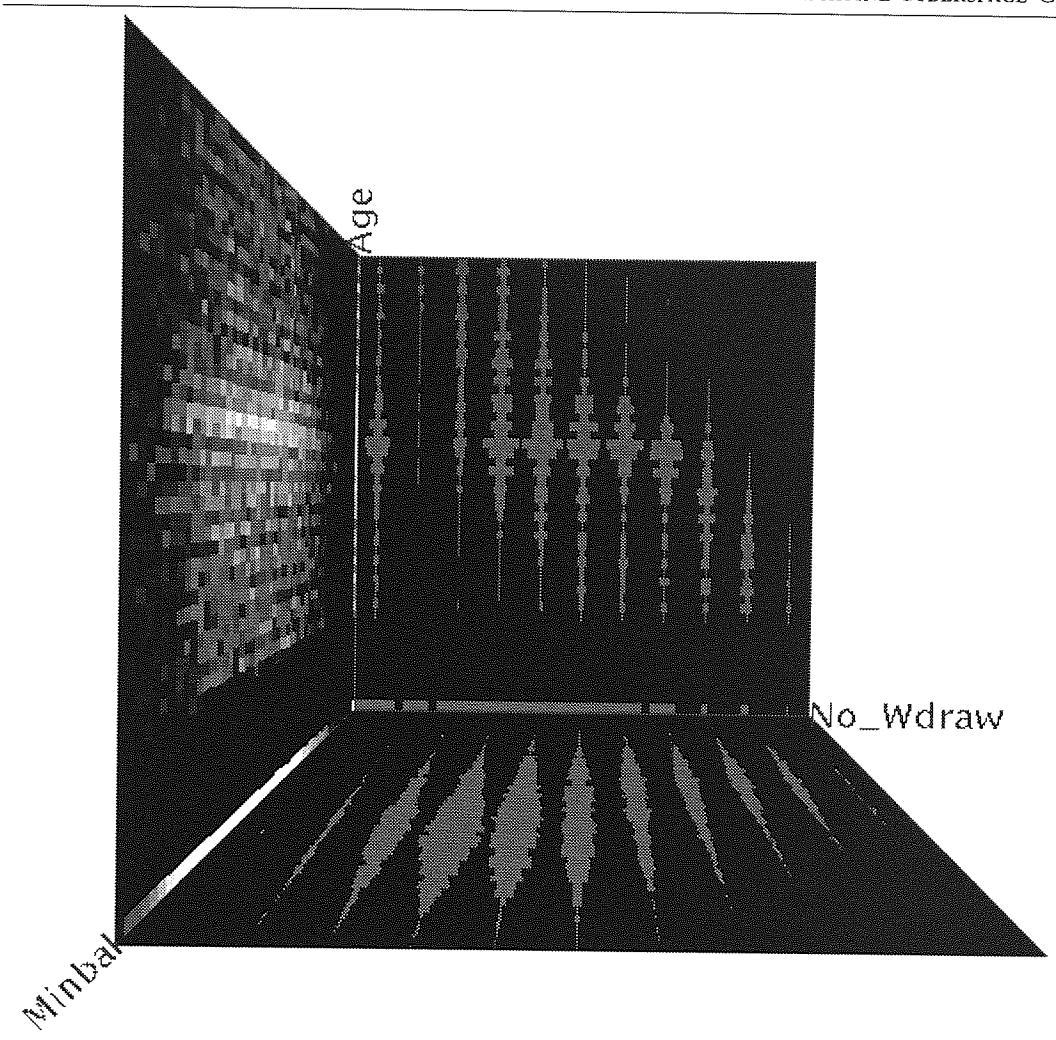
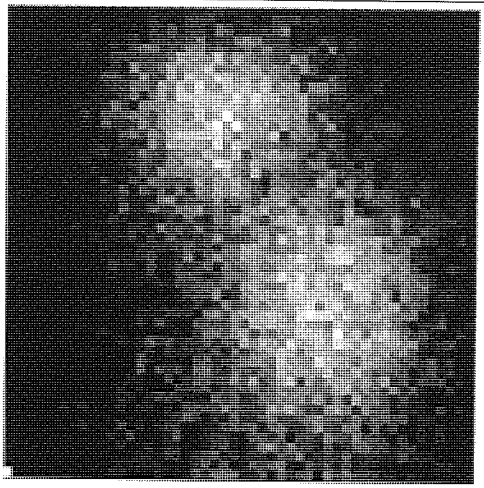


Figure 3.1 – Initial view of the cell (using the mail database)

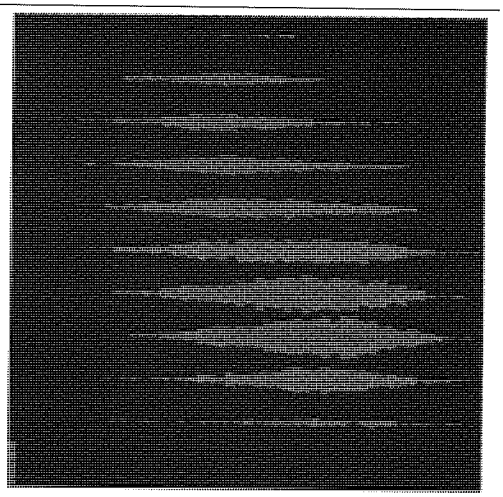
3.3.4.2 Wall display

The walls display either a greyscale image map (for continuous-continuous projections), or the ribbons or rectangles as previously described, in a green colour. Figure 3.2 shows examples of these displays.

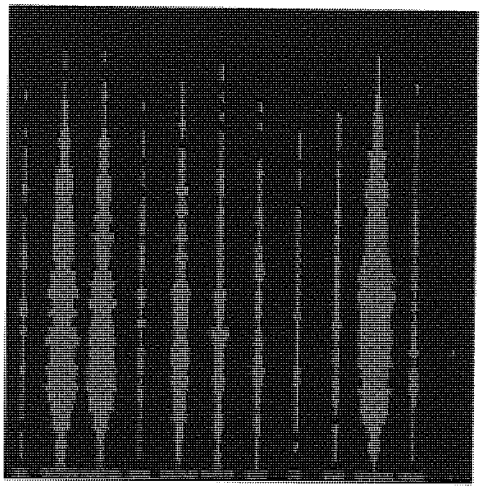
Once the system was capable of showing the walls in 3-D with the data density projected onto them, it proved easy to imagine a 3-D 'cloud' of data hanging in the cell, with its 'shadow' projected onto the three walls.



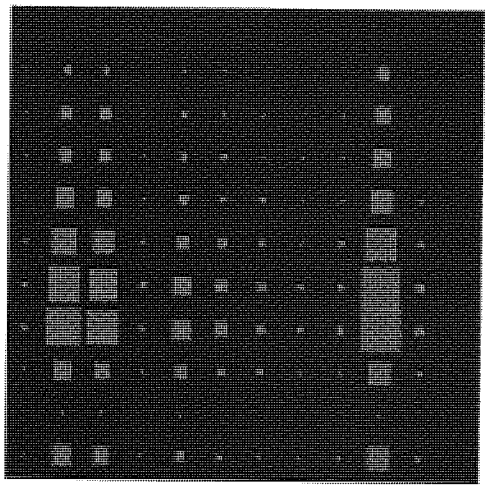
x continuous, *y* continuous



x continuous, *y* discrete



x discrete, *y* continuous



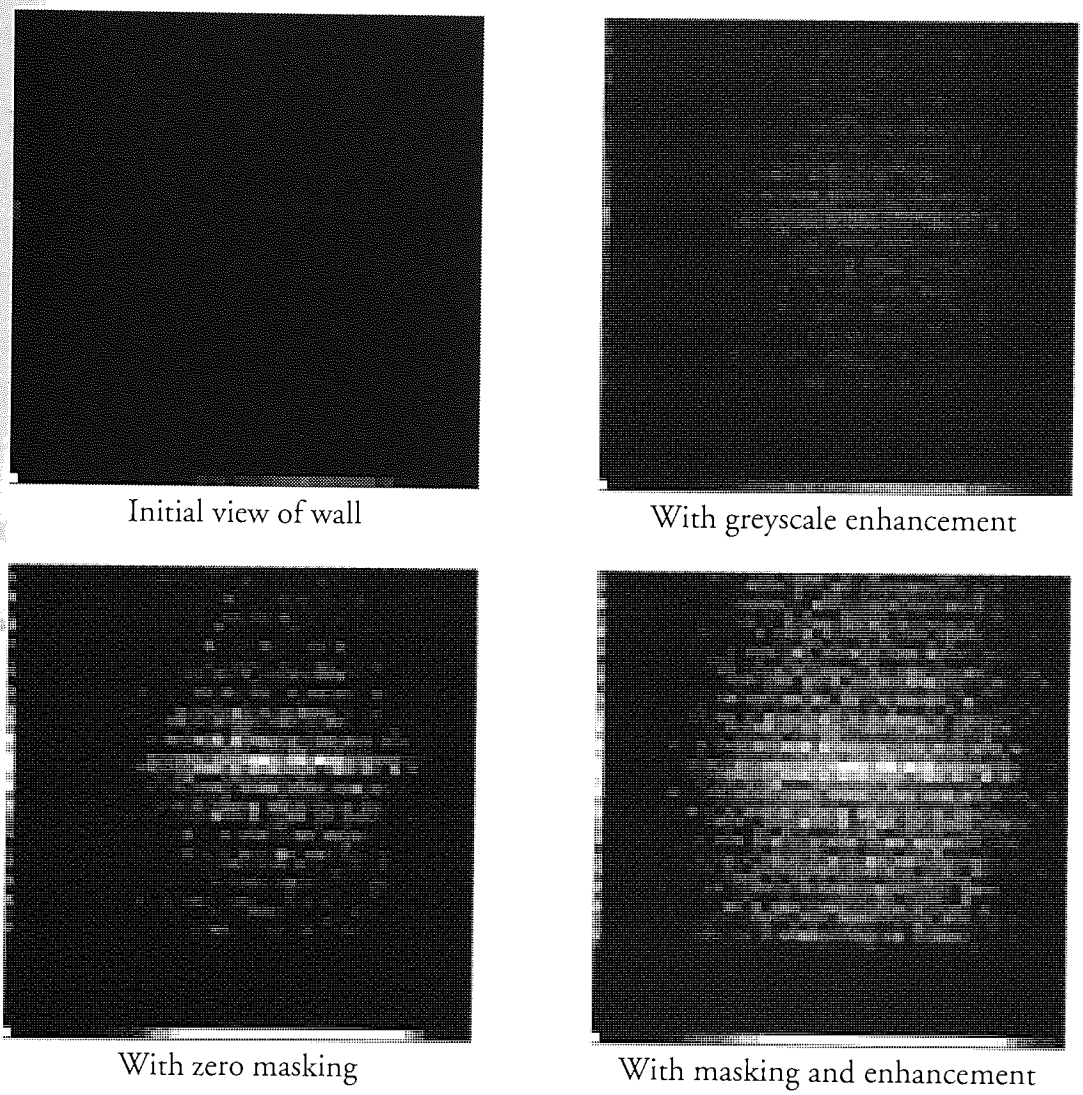
x discrete, *y* discrete

Figure 3.2 – Example of wall displays

Using the system to display real data brought an unexpected problem – since many of the fields in the databases had missing values coded as zero (or a similar low value) or many genuine zero values, the density in the lowest row and column of many density matrices was much higher than elsewhere in the matrix. Due to the 0...1 linear scaling, the majority of the wall was very dark, with a bright white area close to the axis. The problem was worsened when two fields with high 'zero density' were used in the same cell, resulting in a very bright patch at the origin, and very dark areas elsewhere. In the case of ribbons or rectangles displays, one section of the ribbon, or one rectangle, was very large and the rest were very small.

This problem was tackled in two ways. Firstly, the colour scale was made quadratic, by taking the square root of the density value before indexing into the linear colour scale. This has the effect of 'enhancing' the low- and mid-density areas by reducing the resolution in the high-density areas. Secondly, the density projection routines were modified to allow the first row and column of the density matrix to be ignored

when searching for the maximum value required to accomplish the 0...1 scaling. Any entries in the matrix which exceeded 1 after the scaling were clipped to 1. The results of these measures can be seen in figure 3.3. The user was given control of both enhancement and zero masking on each wall. Enhancement of ribbons or rectangles appeared to be a useful feature, so the option to enhance all wall types was made available (shown in figure 3.4 on page 68).



Initial view of wall

With greyscale enhancement

With zero masking

With masking and enhancement

Figure 3.3 – Solutions to high-density areas along the axes.

3.3.4.3 Axis labels

In a traditional graph, there are two labels on each axis: the name of the variable plotted on that axis, and some indication of the values at specific points along the axis, usually including the minimum and maximum values. Initially, all labelling of the cell axes was displayed in a separate information window. However, it was felt that this did not offer an intuitive method of seeing which fields were assigned to the three axes displayed on screen. In order to avoid obscuring any data, the labels were

placed on an extension of each axis, starting at the extremities of the joints between the walls, in clear yellow text, as seen in figure 3.1. The text was rotated to align with the axis, using the public domain *xvertext* library.

No suitable method was found for providing value labels on the axes, so a separate window was used, shown later in figure 3.9 (page 75).

3.3.4.4 Initial choice of fields

The initial choice of three fields to assign to the three cell axes was made on the basis of the standard deviations of the fields. The field with the highest standard deviation was assigned to the x axis, the second highest to the y axis and the third to the z axis. This method favours widely-spread fields, which may be of more interest to the user than fields where there is little variation in value. A sophisticated method for initial field choice will be discussed in chapter 4.

3.3.5 Overlays

It was realised early on during the development of this system that a *fourth* dimension of the database could quite easily be visualised by using a coloured 'overlay' on the walls. In particular, this would be of great use when visualising customer databases with a 'response' field – with the aim of highlighting the areas of the database containing customers who respond more than in other areas[2].

The projection routines were modified to return two matrices, one being the density of the entire database as before, the other being the density of the overlay field projected onto the same axes.

Specifically, the overlay density matrix is generated in a similar way to the density matrix. The matrix elements are incremented by the value of the overlay field in the record under consideration, rather than by unity. Once all records have been processed, each element of the overlay density matrix is divided by the corresponding element of the density matrix (ignoring elements with zero density) to give an average overlay value at each location on the wall. Finally, the matrix is scaled by subtracting the minimum value of the overlay field and dividing by its range. It is not scaled to 0...1

[2]Since this is the main use for overlays, the term 'response field' will generally refer not only to the field of the database containing the response, but also to the field currently being overlaid.

since this would generate the false impression that the locations with the highest average response had in fact a 100% response.

The display of the overlay information was accomplished by using a blue-red colour scale (blue representing 'cold' customers who responded negatively, and red the positive responders). The ribbons and rectangles displays use this colour scale directly, but the image maps required a little more thought. The density information still had to be displayed in some form, otherwise the wall would simply show the average overlay value. The solution was to use a set of four blue-red colour scales, with four levels of brightness. The walls then display the overlay value and the density simultaneously, albeit with reduced density resolution.

Examples of the use of overlays are shown in plates 3.1, 3.2 and 3.3. Plate 3.3 is interesting as it has the overlay field set to the same field as the x axis. This clearly shows the blue-red colour scale.

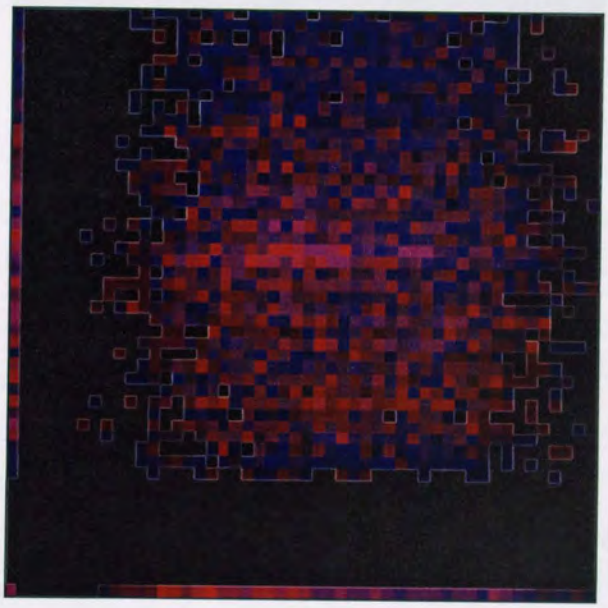


Plate 3.1 – Appearance of the overlay on a continuous/continuous wall

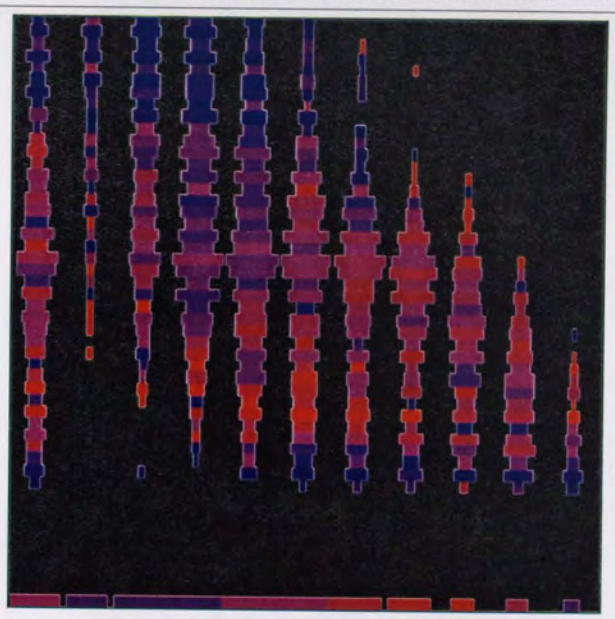


Plate 3.2 – Appearance of the overlay on a continuous/discrete wall

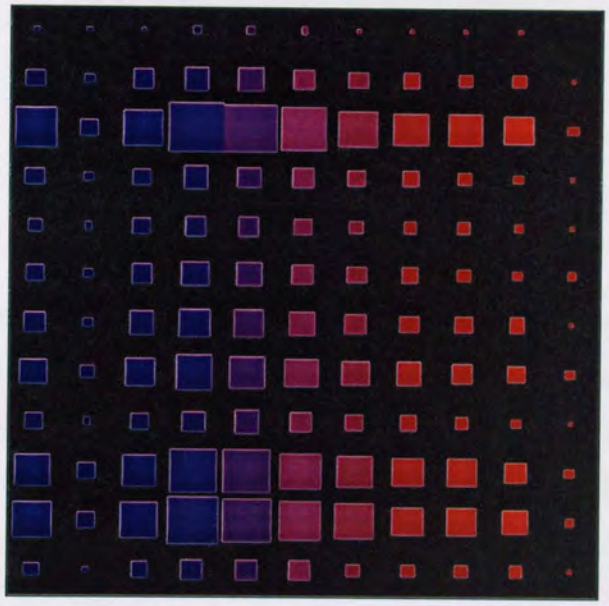


Plate 3.3 – Appearance of the overlay on a discrete/discrete wall, with the overlay field the same as the *x* axis field

Since the usual choice of overlay field in all the test databases was the final field, the overlay was initially set to the final field, though it can of course be changed, as discussed in the following section. The user can enable and disable the display of the overlay on each wall using the control window shown in figure 3.4.

Because of the impossibility of conveying the varying-brightness colour scale in a greyscale picture, all figures in this thesis which are not printed in colour have the overlay display disabled.

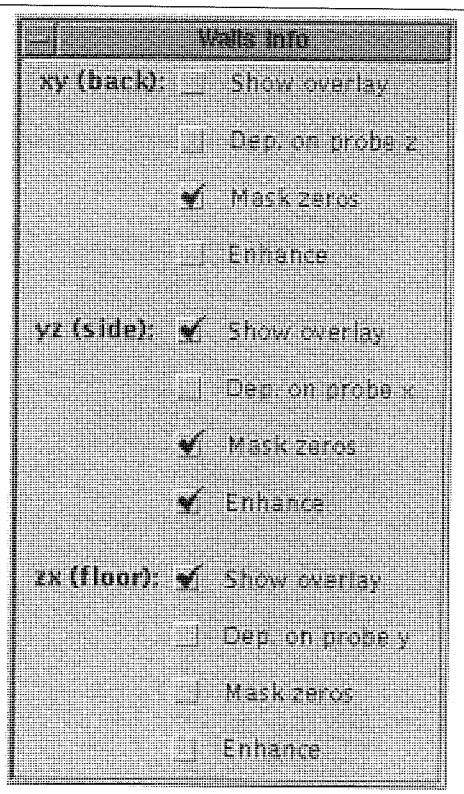


Figure 3.4 – Walls control panel

3.3.6 Axis selection

Several methods of choosing fields to assign to the three axes and the overlay were investigated, including scrolling lists, pull-down menus and arrays of buttons. The technique which gave the best user interface was to use a pop-up menu activated by pressing the right mouse button anywhere on the displayed view. This menu has four sub-menus (one for each axis and one for the overlay) which list all the field names, with the current choice highlighted, as shown in figure 3.5 overleaf.

As soon as a new field is chosen, the affected walls (two for an axis field, up to three for the overlay field) are recalculated and redrawn.

3.3.7 Vehicle movement

Again, several methods of moving the 'vehicle' from which the user is conceptually viewing the cell were examined. As before, the most intuitive involved using the mouse on the displayed view. The left button is used to move in the plane parallel to the screen (up, down, left and right); the middle button allows movement forward and backwards and rotation to the left and right about a vertical axis (yaw). Roll and

pitch (rotation around the two horizontal axes into and across the screen) were not implemented, as it would have made the 3-D projection excessively complex, required a more complicated user interface, and probably disorientated the user.

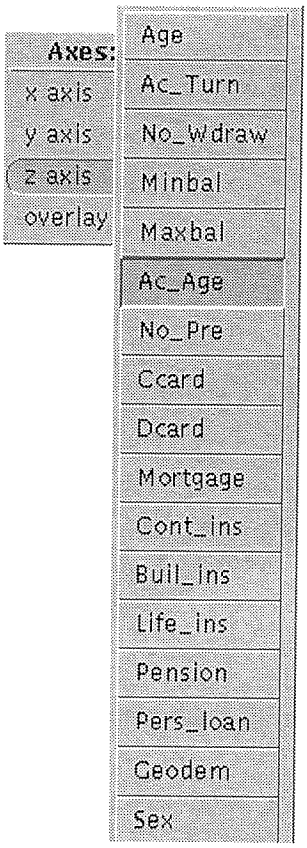


Figure 3.5 – Axis selection popup menu (truncated)

3.3.8 Vehicle movement

Again, several methods of moving the 'vehicle' from which the user is conceptually viewing the cell were examined. As before, the most intuitive involved using the mouse on the displayed view. The left button is used to move in the plane parallel to the screen (up, down, left and right); the middle button allows movement forward and backwards and rotation to the left and right about a vertical axis (yaw). Roll and pitch (rotation around the two horizontal axes into and across the screen) were not implemented, as it would have made the 3-D projection excessively complex, required a more complicated user interface, and probably disorientated the user.

A result of repositioning the vehicle in the initial cell of the mail database is shown in figure 3.6. As can be seen, the viewpoint lies some way outside the cell. This goes against Benedikt's 'implicit vehicle position envelope' which constrains the vehicle to lie within an envelope extending a little way beyond the cell walls. Such a constraint

was implemented but it was felt that free movement of the vehicle was preferable, particularly in order to achieve 'long distance' views such as figure 3.6.

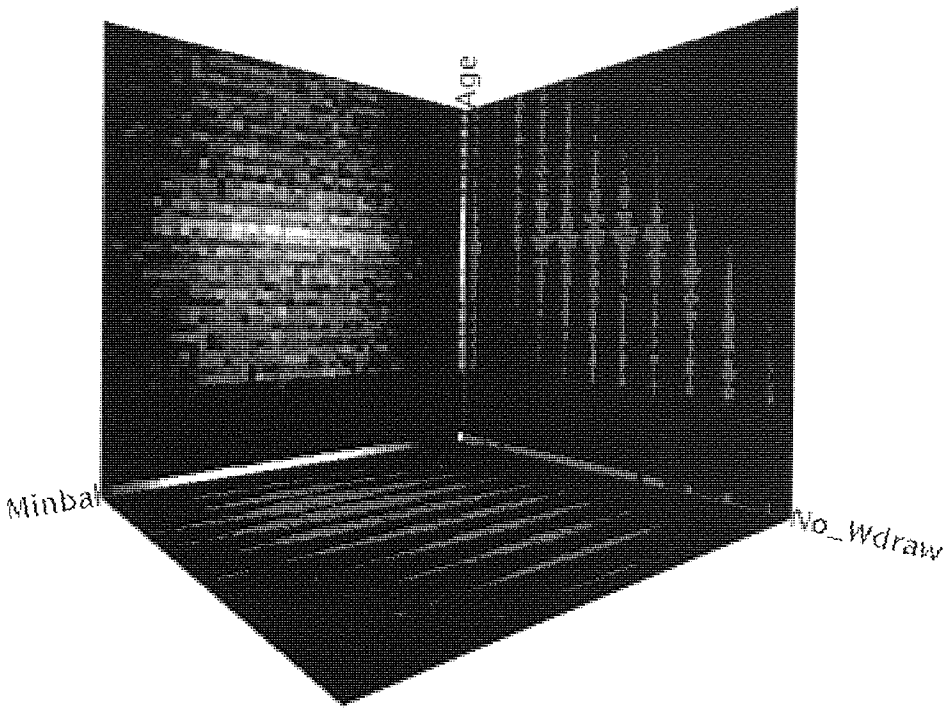


Figure 3.6 – View after moving and turning the vehicle

All mouse clicks for vehicle movement are made relative to the centre of the view: clicking above the centre moves up or forward; below the centre moves down or backwards; left of centre moves or turns left; right of centre moves or turns right. The distance moved, or angle turned, is proportional to the distance from the centre, and two movements can be combined – clicking the left button in the top left of the view moves up and left; the middle button in the top right turns right then moves forward.

Following testing, some options were added to the movement controls:

- Rotation can be disabled, for example to prevent the view skewing when the user is moving towards a perpendicular wall.
- The up/down and forward/backward functions can be swapped, making the left button control movement left/right and forward/backward and the middle button control rotation and up/down movement.
- An optional crosshair can be displayed to aid positioning of mouse clicks.

These options are grouped onto a popup properties window, as shown in figure 3.7.

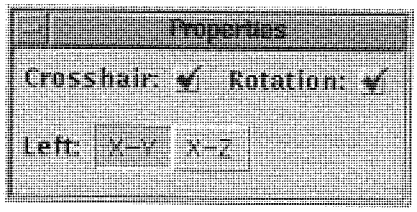


Figure 3.7 – Properties control panel

On a machine optimised for fast 3-D graphics performance, it might have been possible to implement continuous movement so the user could hold the mouse button down and interactively move around the cell. Unfortunately the available hardware precluded such effects, and there is a short delay after pressing the button before the display updates. The update is smooth however, with a double-buffering technique allowing the current view to remain visible until the new view is ready to be shown.

3.3.9 The probe

Benedikt's description of the cell calls for a 'probe' which can be moved to select a 3-D location in the cell. This selection has two functions. Firstly, each wall can be set to display the projection onto its two axes given the currently selected value of the other axis (see section 3.3.10), and secondly, the selected records of the database can be used to create a new cell via unfolding (see section 3.3.11).

3.3.9.1 Probe size

There were several options for the size of the probe:

- A point, selecting one value for discrete fields, and one precise value for continuous fields.

  This created problems with continuous fields – since only a precise value is selected, it nearly always selects a value for which there is no data.

- An 'element', selecting one density matrix element along each axis, encompassing a range of values (one 'bin') for continuous fields.

  This method gave much improved performance, selecting exactly what can be seen on the walls of the cell. However the constraint on continuous fields was again a problem: with an 'age' field with values 0...70 'binned' into fifty bins, each element has to be of width 1.6 years, which is unintuitive.

- A set of these 'elements', selecting one or more *contiguous* locations on the axes, again encompassing a set of 'bins' for continuous fields.

  The move to larger selections was a great benefit. Now ranges could be selected and selection sizes adjusted to fit features observable on the walls. The use of bins on the continuous fields was still problematic though.

- A variable-size cuboid, capable of selecting any precise range of continuous fields, but naturally constrained to integer values for discrete fields.

  This solution is the one which was implemented in the final system. It has the same benefits as the previous technique, but overcomes the limitation on continuous selection by allowing an exact range to be specified.

It might have been possible to envisage a discontinuous or non-rectangular selection with an advanced probe, but this was considered impractical and difficult to implement in 3-D.

3.3.9.2 Probe display

Evidently the method of displaying the position and size of the probe in the cell was very important. It had to clearly convey the location and extent of the selection while not obscuring the view of the wall displays.

Initial experiments with methods for showing a point selection will be not be detailed here, since a probe of variable size was chosen for the final system. Some of the ideas for displaying the probe are shown in figure 3.8 on page 74:

- A cuboid hanging in the cell, enclosing the space where the 'data cloud' selection lies, as shown in figure 3.8*a*. This was rejected as it was unclear to the user exactly where the six boundaries of the selection were.

- The same cuboid, with its twelve edges extended to intersect the walls, as shown in figure 3.8*b*. This helped to locate the selection in space, but the effect was still unclear.

- As above, but with the four intersections on each wall joined to form the 2-D selection rectangle, as shown in figure 3.8*c*. This gave a much clearer indication of the selection, but the cuboid tended to confuse the display and obscure data behind it.

- Using only the three rectangles on the walls, leaving nothing 'in' the cell at all, as shown in figure 3.8*d*. This 'minimalist' display was very clear. However, when used with dependent axes (see section 3.3.10 below) it was not clear which data points were selected on a particular axis (e.g. the range of just the x selection, rather than the x-y and x-z combined selection rectangles).

- As above, but with the four edges of the rectangle on each wall extended to the edges of the wall, as shown in figure 3.8*e*. This clearly showed which data points were selected on each axis but was rather too 'intrusive' for continuous use.

The final system uses rectangles on the walls for normal selections, changing to the extended form described above only when the axis is being used to control a dependent wall. Figure 3.8*f* shows the display when the z axis is being used to control the dependent x-y wall.

The probe lines are drawn in yellow, using a third screen buffer, allowing the probe's position and size to be changed without requiring the whole display to be redrawn. The initial position of the probe is at the origin, with its size set to one in each direction (covering one value of discrete fields and a range of unity on continuous fields).

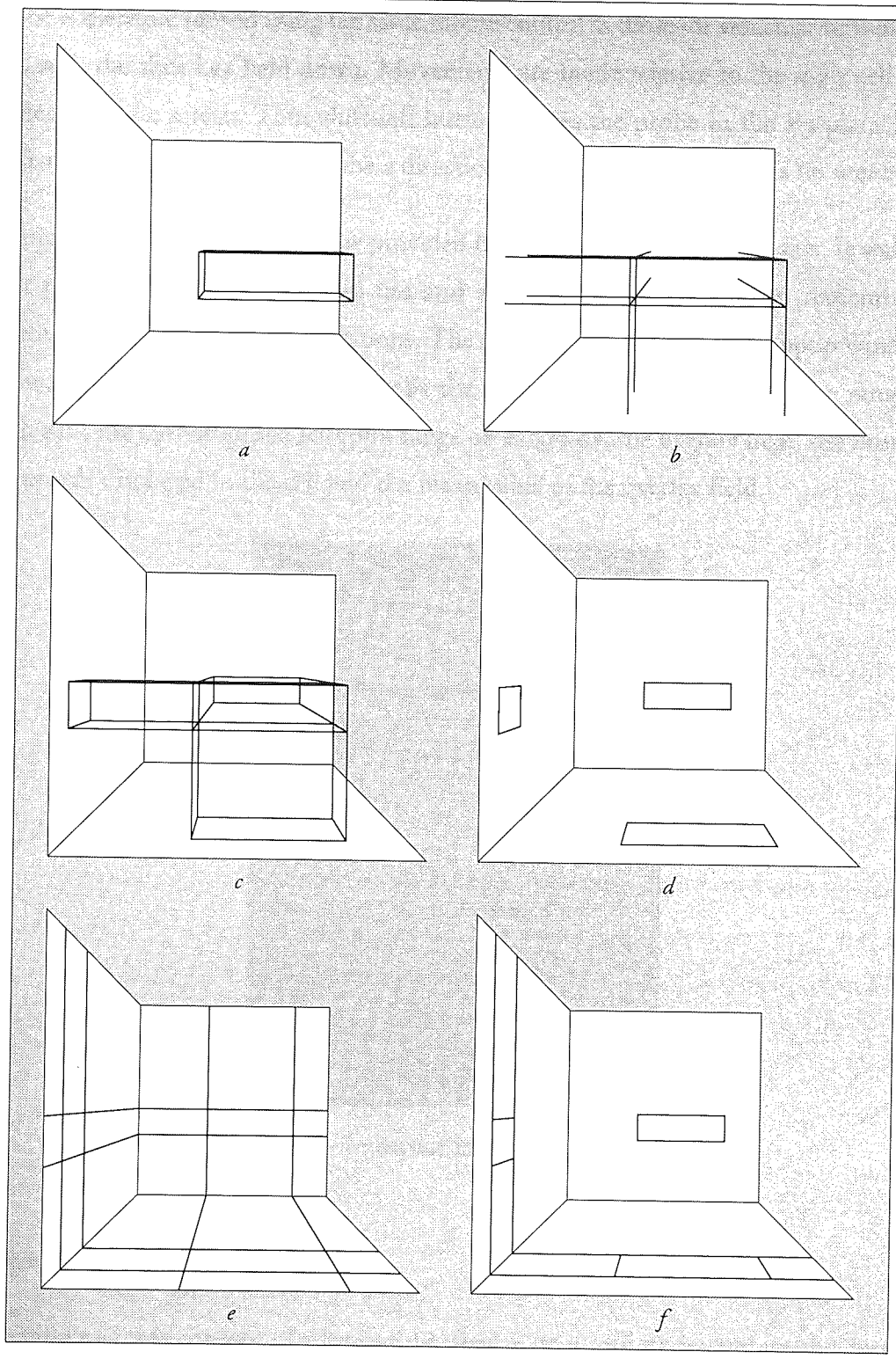


Figure 3.8 – Evolution of the probe display.

3.3.9.3 Probe control

Benedikt demands that the probe be moved by 'clicking and dragging' with a multi-dimensional controller. This would have been difficult to implement using a standard mouse, particularly given the chosen interface method for moving the vehicle. The

probe is therefore moved using the same mouse control as those for vehicular movement, but with the shift key held down. Movements are made relative to the *x-y-z* cell axes rather than the screen. Thus shift-left button moves the probe in the *x-y* plane, and shift-middle button moves it in the *z* direction (rotation of the probe has no meaning).

Control of the size of the probe is provided by a 'width' slider for each axis. It was felt that this interface was simple to use and avoided the user having to remember a further six mouse/keyboard operations. The sliders are contained in a popup window, shown in figure 3.9, which also details the field assigned to each axis, the range of each axis, the current probe selection range on each axis, the overlay field, the number of records displayed in the cell and the mean value of the overlay field.

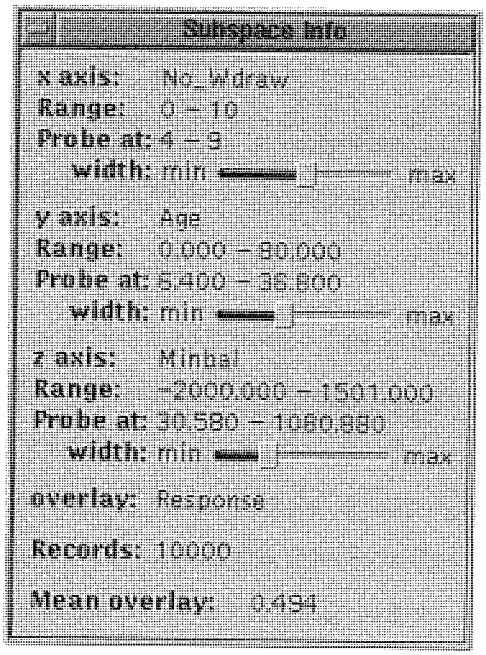


Figure 3.9 – Subspace information window.

3.3.10 Dependent walls

Benedikt's simple concept of allowing the display on a wall to depend on the current selection on the perpendicular axis gives a powerful tool for data analysis. For example, if the *x-y* wall shows a distribution of interest to the user (maybe using an overlay), and the *z* axis is assigned to the 'customer age' field, then by making the *x-y* wall *dependent* on the *z* selection, the user can see the *x-y* density for any selected age range. By moving the probe back and forth in the *z* direction, the *x-y* wall shows the changing density of the moving 'slice' through the database, hopefully revealing previously unseen relationships.

Once more, the projection routines were modified, to project only those records enclosed in the one-dimensional probe selection along the perpendicular axis.

The user has the option to set each wall to be dependent via the 'walls info' window (see figure 3.4). As explained in section 3.3.9.2, the probe display changes to indicate when an axis is being used as a dependency controller.

3.3.11 Subspaces

The final element of the Benediktine cell is the ability to unfold new 3-D spaces from the intrinsic dimensions of selected data object in the existing cell. In the implementation described here, these new spaces are termed *subspaces* since they contain a lower-dimensional subspace of the initial database.

Implementation of subspaces was relatively straightforward, due to the object-oriented nature of the program. Each subspace (including the original cell) is a C++ object which has three wall objects, a probe object, a database object, etc.

The user interface is provided by three buttons: 'open subspace', 'next subspace' and 'delete subspace'. When the 'open subspace' button is pressed, the data objects contained in the current 3-D probe selection are identified and used to create a new database for the new subspace. The new cell is located in the centre of the enclosing cell, and is 75% its size (the user will typically move the vehicle towards the new cell to enlarge its appearance on the screen). The new cell is made 'active', and the outer cell is made 'inactive'. Inactivity is indicated by the cell walls becoming transparent.

The 'next subspace' button cycles through all the subspaces in the system, making each one active in turn. This allows the user to make any desired subspace active. The 'delete subspace' button deletes the currently active subspace, and any subspaces which were created from selections in this space. The initial space cannot be deleted.

Since the user may wish to create more than one subspace at the same hierarchical level (i.e. from selections in the same cell), a means of moving the cells was needed. This was implemented in the same way as probe movement, substituting the control key for the shift key. Thus control-left button moves the active subspace in the x-y plane, etc.

Figure 3.10 shows a display featuring an unfolded subspace. The transparent walls of the outer cell can be seen in the background.

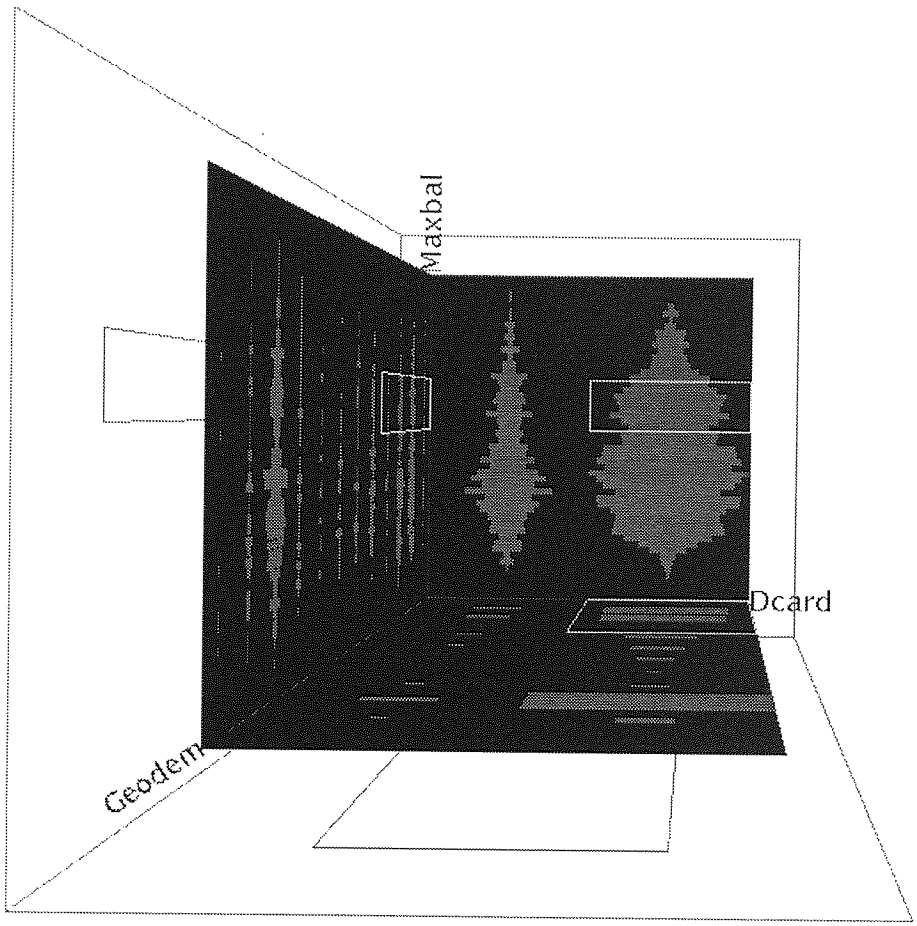


Figure 3.10 – Display showing a subspace

The requirement that the contents of the subspace should change as the selection in the outer cell changes was not implemented, for several reasons:

- The update process to recalculate the database and redraw the inner cell takes a long time, which would frustrate users.
- Under the current system, only one subspace is active at once – it is impossible to manipulate the probe in one space while observing changes in another.
- Using the system with real data (see below), a situation which required this feature did not arise.
- By this point in the development process, emphasis was switching to a new visualisation techniques which avoided the issue. This new system is detailed in the following chapters.

3.4 Use with Real Data

Once the system was complete, it was used to visualise two of the real-world databases described in section 1.2.4 and detailed in the appendix.

3.4.1 Mail database[3]

Using three of the continuous fields, `Age`, `Ac_Turn` and `Minbal`, as an initial assignment to the axes, and moving the viewing position so that the three wall displays are equally visible results in the view shown in figure 3.11.

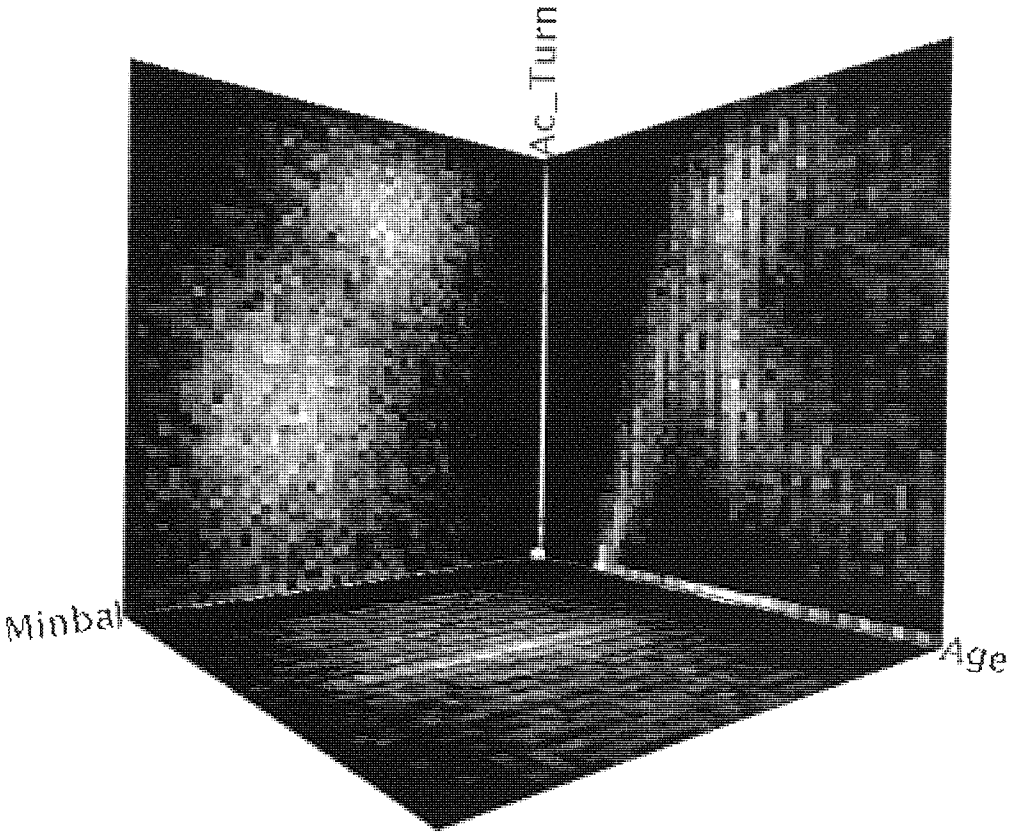


Figure 3.11 – A starting point for exploration of the mail database

Several interesting observations can be made from these three wall displays:

- There are two clear elliptically-shaped groups on the *y-z* wall, one with moderate minimum balance and high turnover, the other with higher minimum balance and lower turnover. This may well reflect accounts being used respectively as current accounts and savings accounts.

[3]Some of the results in this section were presented at the Unicom seminar on Adaptive Computing and Information Processing and subsequently published in the proceedings [Bounds & Barrett, 1994].

- There is a complex relationship between account turnover and customer age. As age increases, the turnover also increases, until approximately age 35 (ascertained by moving the probe), when there is a clear divergence into accounts with high turnover and those with low turnover. As age increases above 65, the turnover drops off, reflecting the circumstances of pensioners.

- Many customers have zero age (shown by the dense strip on the left of the back wall), zero account turnover (the dense strip on the bottom of the back wall), or both (the corner of the back wall). Zero age represents missing information in the database, but zero turnover is likely to be valid, indicating dormant accounts.

If the overlay is now used to show the Response field, the view shown in plate 3.4 results.

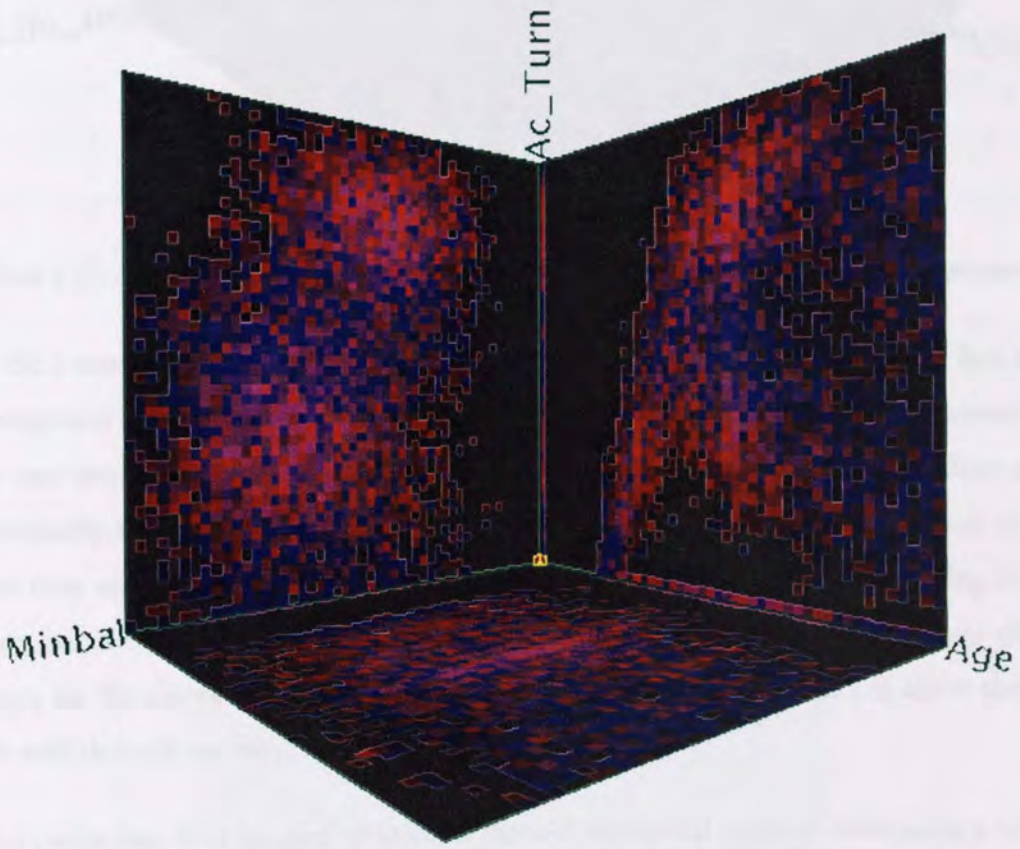


Plate 3.4 – A view of the mail database with the Response field overlaid

The Ac_Turn-Minbal wall shows a relationship between response and account turnover which is similar, but not equal, to the density of the same wall, shown in figure 3.11. Two fairly well-defined bands of response can be seen: one with high turnover, the other with lower turnover. The Ac_Turn-Age wall and the floor clearly show that there is also a well-defined age range where a positive response occurs, approximately

21–66 years. This is intuitive: few people under 21 or over 66 purchase life insurance policies, for different but obvious reasons.

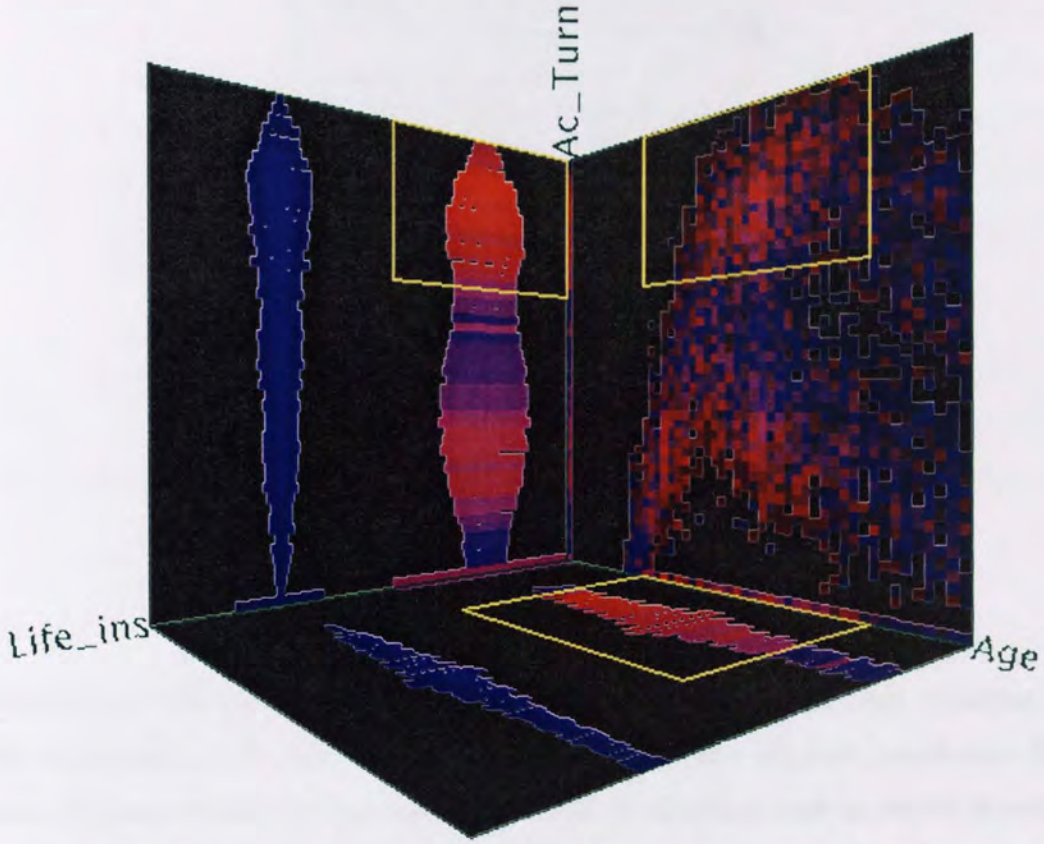


Plate 3.5 – A different view of the mail database, with a selection of likely responders

If the *z* axis is changed to `Life_ins` (as shown in plate 3.5), an immediate but not unexpected result becomes clear: those customers who already have some life insurance do not respond to the mail shot offering additional life insurance. This does not necessarily mean that these customers are less likely to purchase life insurance, only that they are less likely to respond to cold mail shots; they may well be making their own insurance arrangements pro-actively rather than responding to whatever offer drops on the doormat. The `Life_ins-Age` wall (the floor in plate 3.5) again shows the well-defined age range.

The probe can now be used to select a three-dimensional volume containing a high proportion of respondents, as shown in plate 3.5. The selection encompasses customers with ages 21–66, no other life insurance, and account turnovers of £1274–1930 per month (as shown in the subspace information window, figure 3.12).

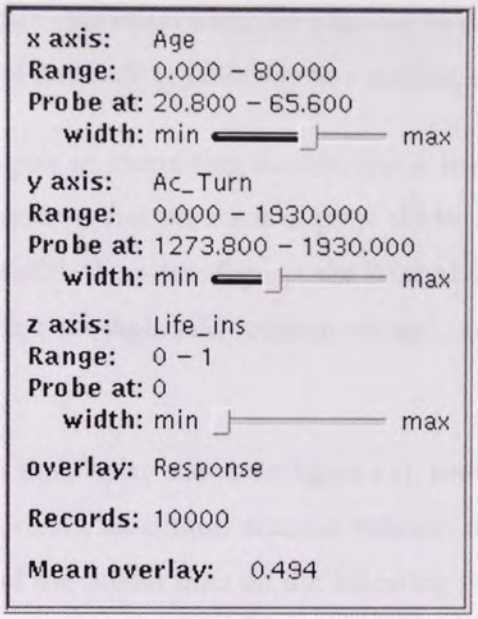


Figure 3.12 – Subspace information window showing details of the selection of high respondents

A subspace containing only the selected customers can now be opened, resulting in the display shown in plate 3.6. The subspace information window reveals that this new subspace contains 1520 customers (15.2% of the database) with an 85.7% response rate (compared with 49.4% in the entire database).

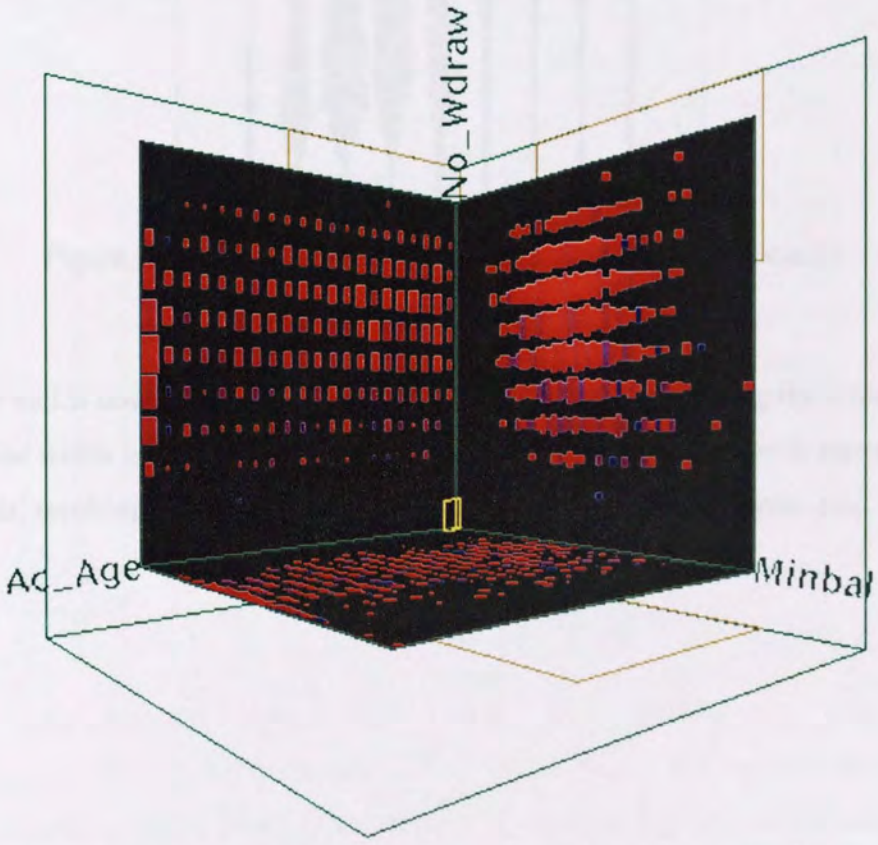


Plate 3.6 – The subspace opened from the selection shown in plate 3.5

In this way, visual information has been used to define subsections of three fields of the database which contain customers who, the response to the test mailing indicates, have a high probability of responding positively to a mailing offering life insurance.

Next, the effect of age upon an interesting distribution is investigated. The subspace is deleted, and the axes reset so that the *x* axis displays the `No_Wdraw` field (number of ATM withdrawals per month), the *y* axis displays the `Maxbal` field (maximum account balance) the *z* axis displays the `Age` field (customer's age), and the overlay is turned off.

The *x*-*y* wall now shows the display shown in figure 3.13, showing the distribution of number of withdrawals versus maximum account balance, over the entire database. Note that this figure, and the similar ones on the following pages, have been inverted for clarity when printed.

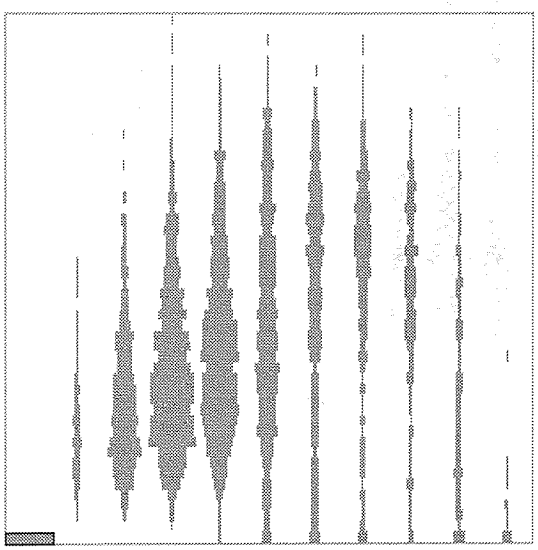
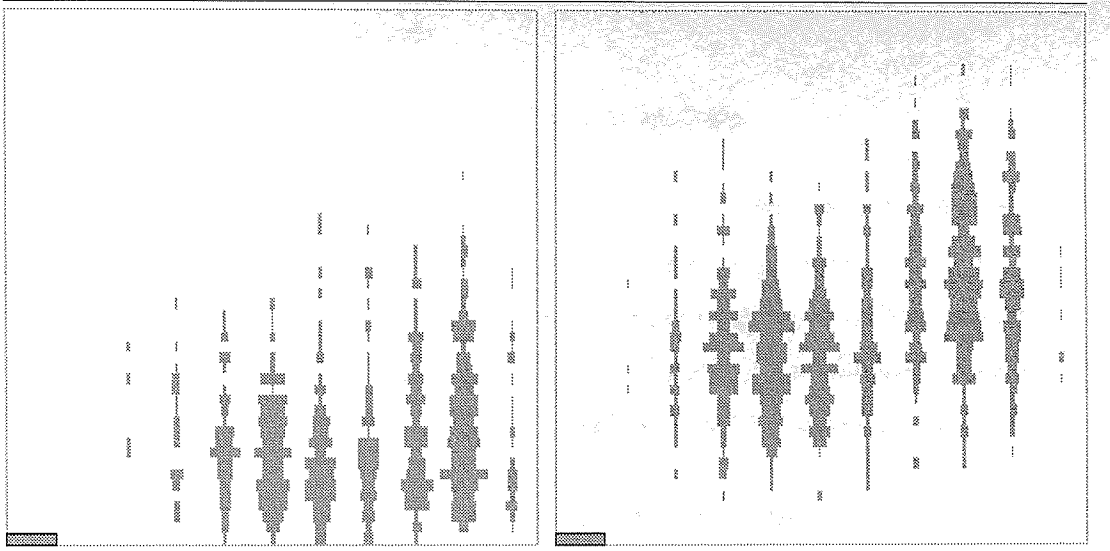
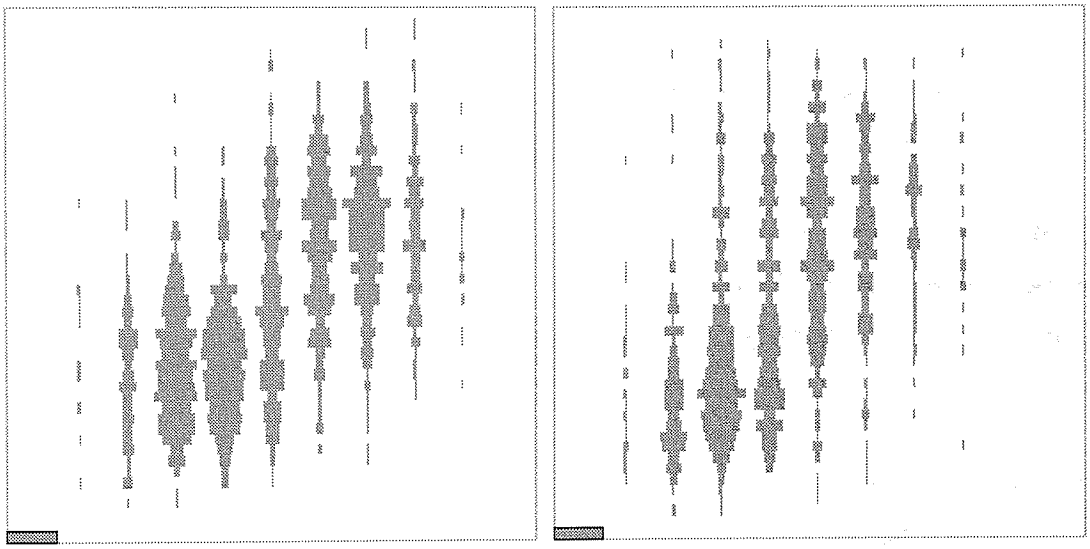


Figure 3.13 – Wall showing number of ATM withdrawals (*x* axis) against maximum balance (*y* axis)

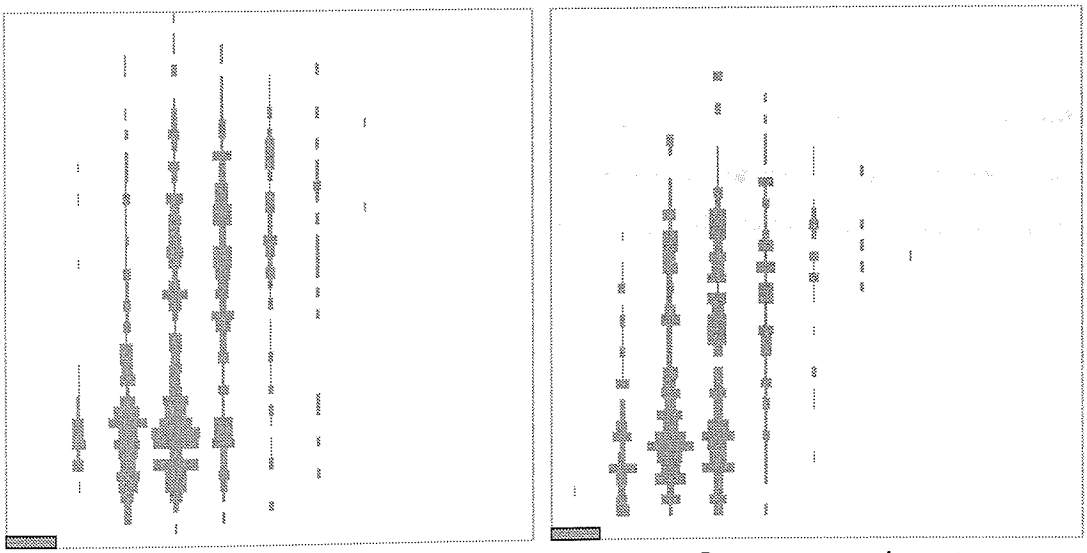
The *x*-*y* wall is now made dependent upon the probe selection along the *z* (`Age`) axis, the probe width is set to ten years in the *z* direction, and the probe is moved along the *z* axis, resulting in the series of wall displays shown overleaf in figure 3.14.



Customers aged 20–30 Customers aged 30–40



Customers aged 40–50 Customers aged 50–60



Customers aged 60–70 Customers aged 70–80

Figure 3.14 – Wall showing number of ATM withdrawals (x axis) against maximum balance (y axis) dependent on selection of customer age (z axis, not seen)

Examining figure 3.14 reveals that:

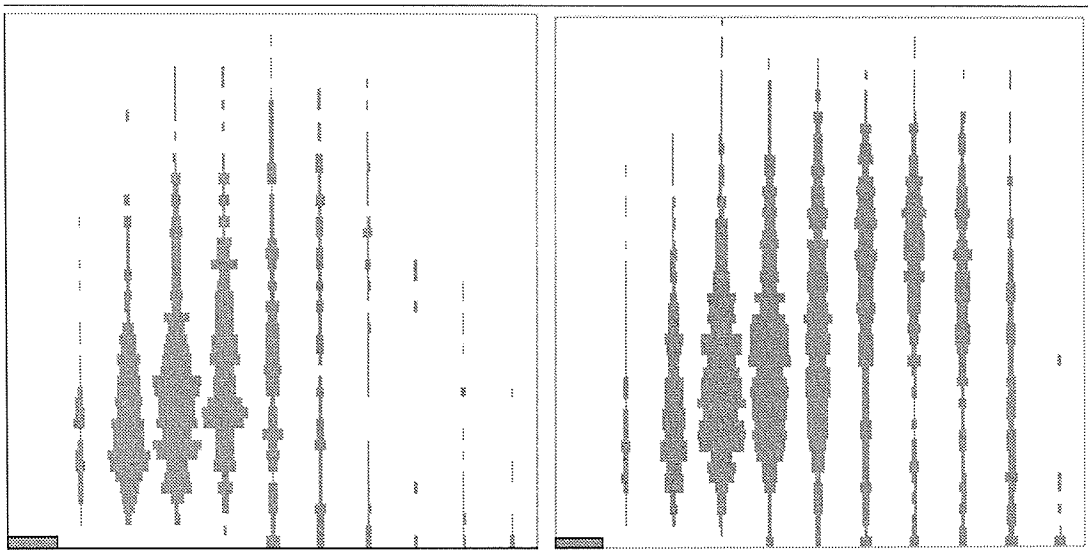
- younger customers make most use of the ATM facilities.
- If `Maxbal` is zero (or technically if it's in the first bin), then `No_Wdraw` is generally zero, probably indicating an empty and inactive account. Exceptions to this observation lie in the plot for customers aged 20–30, where some accounts with zero or near-zero maximum balance have withdrawals noted. This is quite possibly due to students struggling to survive on their overdrafts.
- All customers with non-zero maximum balances make at least one ATM withdrawal, with the exception of a very few customers in the 70–80 age range.

All these findings would be of use to a marketing manager investigating the database, who would probably be able to discover more, using background knowledge.

If the probe is moved (and resized) to select only those customers whose age is recorded as zero (i.e. whose age is unknown), the rear wall shows the display shown in figure 3.15*a*. Changing the selection once more, to include all ages except zero, results in figure 3.15*b*.

Comparing figure 3.15*a* with figures 3.14 and 3.15*b*, it seems clear that it bears a close relationship to figure 3.15*b*. Thus it may be surmised that the real ages of customers whose age is unknown are distributed throughout the age range with a similar distribution to that of customers of known age.

There is one clear difference between figures 3.15*a* and 3.15*b* : there are considerably more customers who make more than 6 ATM withdrawals whose age is known than whose age is unknown. This discovery might well prompt further investigation by the user investigating this database.



a – Age zero *b* – Age non-zero

Figure 3.15 – Walls showing number of ATM withdrawals (*x* axis) against maximum balance (*y* axis) for customer age zero, i.e. where the age is unknown and for non-zero customer age, i.e. where the age is known

3.4.2 RAE database

Visualisation of the RAE database using the Benediktine visualiser revealed little useful information, mainly due to the fact that virtually all the fields in the database are skewed – the majority of records lie within a small range, but a few records have very large values. Thus most of the 2-D wall projections look similar to figure 3.16 (in fact, this is one of the most informative).

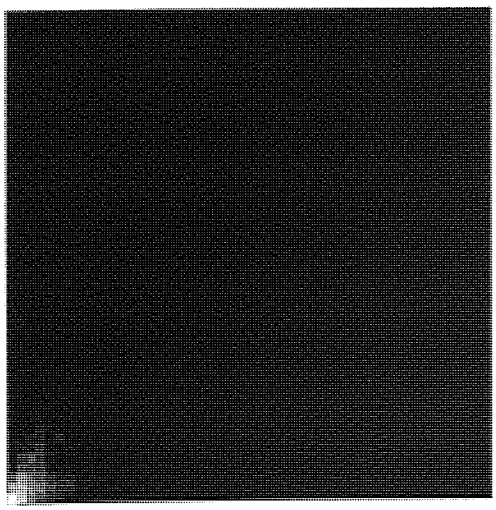


Figure 3.16 – Wall showing `inpost` against `pub1` from the RAE database

One projection which does give some insight into the rating awards is the plot of `sel_staff`, the number of research staff selected to contribute to the assessment, against `rating`, as shown in figure 3.17. This shows that, in general, departments with a larger number of selected staff tended to be awarded higher research ratings.

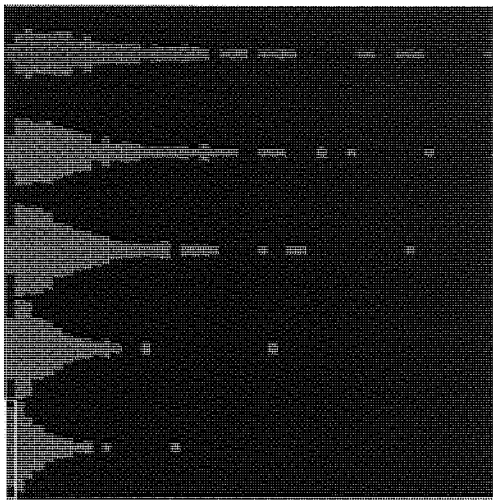


Figure 3.17 – Wall showing `sel_staff` against `rating` from the RAE database

Beyond this, little information could be extracted from the RAE database using the cell visualiser. However, a lot more remained to be discovered, as will be seen in later chapters.

3.5 Conclusions

3.5.1 Concept

The implementation described in this chapter has proved that the Benediktine cell is a powerful visualisation tool which allows users to gain insights into their data which would otherwise have been difficult to discover.

The wall displays allow density projections of both continuous and discrete fields to be easily observed. The extension to show a fourth dimensions using a coloured overlay allows additional information about the data distribution to be presented, although this does have a detrimental effect on the clarity of the original density information.

The 3-D perspective display helps the user to imagine a 'data cloud' hanging in the cell, and by allowing the viewpoint to be moved around the cell, the display can be changed from a long-distance overview to a close-up of an interesting part of a wall.

Dependent walls allow inter-relationships between fields to be investigated by selecting and projecting slices through the cell, as graphically demonstrated by the investigation of the mail database seen earlier.

The creation of subspaces by unfolding three intrinsic dimensions from selected data is a novel technique which allows high-dimensional databases to be explored, three (or four) dimensions at a time.

3.5.2 Implementation

The system as implemented, while definitely proving that Benedikt's idea has considerable merit, has numerous practical limitations.

The low resolution of continuous fields (fifty bins) enforced by speed constraints prevents detailed analysis of density patterns. Experiments with increasing the resolution gave clearer patterns, but took much longer to generate and display.

The lack of value labels on the axes makes it difficult to discover the location of particular features on the walls. In addition, the distortion due to the perspective projection can hinder interpretation of the wall displays.

When the system is running on a modestly-powered workstation, the display takes a long time to update, due to the large number of calculations required to generate the 3-D projection onto the 2-D screen. This problem could be tackled in several ways: by tightly optimising the display code, by using a ready-made 3-D graphics library, or by using a dedicated 3-D graphics hardware accelerator.

The cuboid probe, though restrictive (it cannot, for example, select a cylindrical or spherical cluster of records), could not be made more flexible without considerable effort. Nevertheless, it is powerful enough to allow useful selections to be made.

The popup menu for field selection works well, but the separate window to control overlays, dependencies, zero masking and enhancement is confusing and offers the user too many choices. The system would be a lot simpler to use if some of these options (particularly enhancement and zero masking) were permanently enabled or disabled, or maybe put on an 'advanced controls' window.

The interface to the vehicle and probe movement, using mouse clicks relative to the centre of the screen, is adequate to navigate to a desired view and position the probe, albeit in a rather trial-and-error manner. However, the three sliders which control the size of the probe are decidedly difficult to use when attempting to enclose an area of interest on a wall, though they have an evident advantage for making a selection of a precise size. A better solution would be to implement a click-and-drag interface for adjusting the probe, though this would require back-projection from the 2-D screen into the 3-D cell space.

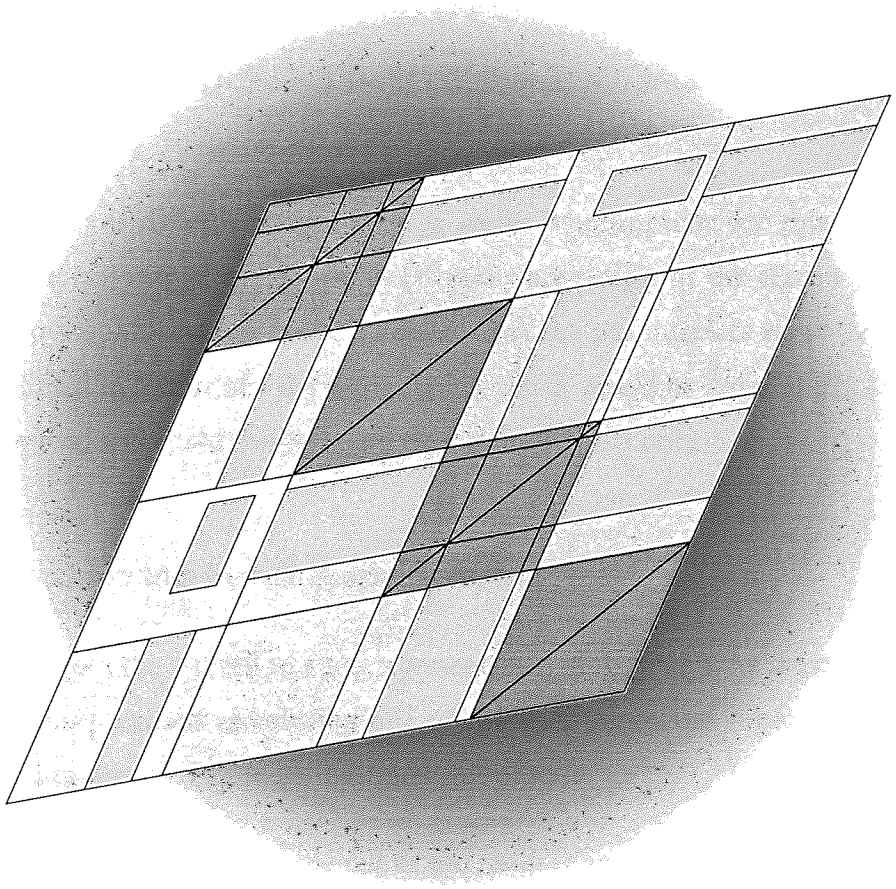
Because of the slow response and the relatively clumsy interface, users quickly become frustrated with the system. On a faster machine, with dedicated graphics hardware, maybe including a stereoscopic display and a specialised 3-D input device, the cell would be a revolutionary and genuinely useful tool.

3.5.3 Practical applications

This cell visualiser system proved to be of use in genetic algorithm (GA) research [Price, 1996]. By assigning 'population index' to the x axis, 'fitness' to the y axis and 'generation number' to the z axis, the cell was used to examine the evolution of a population over time, with dependant walls being used to pick out particular generations or individuals.

Recognition Systems have also demonstrated the system to several of their clients, and report considerable interest. The company is currently porting the code to a Windows platform, with the aim of incorporating it into their commercial data analysis tools.

Chapter 4



Visualisation Tools II: Maden

On ne voit bien qu'avec le cœur. L'essentiel est invisible pour les yeux – It is only with the heart that one can see rightly; what is essential is invisible to the eye.
[de Saint-Exupéry, 1943]

4.1 Concept

As we have seen, the three-dimensional nature of the cell visualiser described in the previous chapter forces three major constraints:

- Only three fields can be visualised at once
- The 3-D projection distorts the data displayed on the 2-D walls
- The 3-D projection requires a lot of calculations which make the system very slow to use

Though the 3-D concept is appealing, it seems inappropriate for this application. Indeed, the majority of the figures in chapter 3 are simply of the rear (*x-y*) wall and require no 3-D information at all. It appears that the 3-D interface must (reluctantly) be abandoned in favour of a flat 2-D interface which would be faster to display, much clearer to read and could display more than three fields at once.

4.1.1 Transformation to a two-dimensional matrix

The move from the 3-D cell to a 2-D representation of (initially) the same information can be conceptualised as splitting one of the wall joints and 'unfolding' the wall structure into its net. Figure 4.1 illustrates this process.

The orientation of the axes in the cell would result in inconsistent axis directions after the unfolding, so it is necessary to reverse the direction of the *z* axis (figure 4.1*b*). Also, in order that the sequence of the axes in the 2-D visualisation be correct (i.e. *x*, *y*, *z*), the *y-z* wall is swapped to the opposite side of the cell (figure 4.1*c*) before splitting the vertical wall joint and unfolding (figure 4.1*d*).

As figure 4.1*d* shows, the result of unfolding the cell is a triangular (sub-diagonal) matrix of Benediktine walls. Each of the three walls contains a 2-D projection, with its vertical axis defined by the *row* of the matrix in which it is located, and its horizontal axis defined by the *column* of the matrix. Extrapolating this definition to the diagonal elements of the matrix would result in walls with identical horizontal and vertical axes, which can therefore be replaced by an *identifier* representing the name of the axis in question (figure 4.1*e*).

The super-diagonal walls can be constructed in the same way, using the row and column indices to determine their axes (figure 4.1*f*). The super- and sub-diagonal walls then present the same information, reflected about a 45° axis. The seemingly redundant plots allow a viewer to look for patterns along rows and columns, without having to 'turn the corner' – for example, when examining the *y* axis, the *x*–*y* and *z*-*y* plots are separated in figure 4.1*e*, but can be seen in one row in figure 4.1*f*.

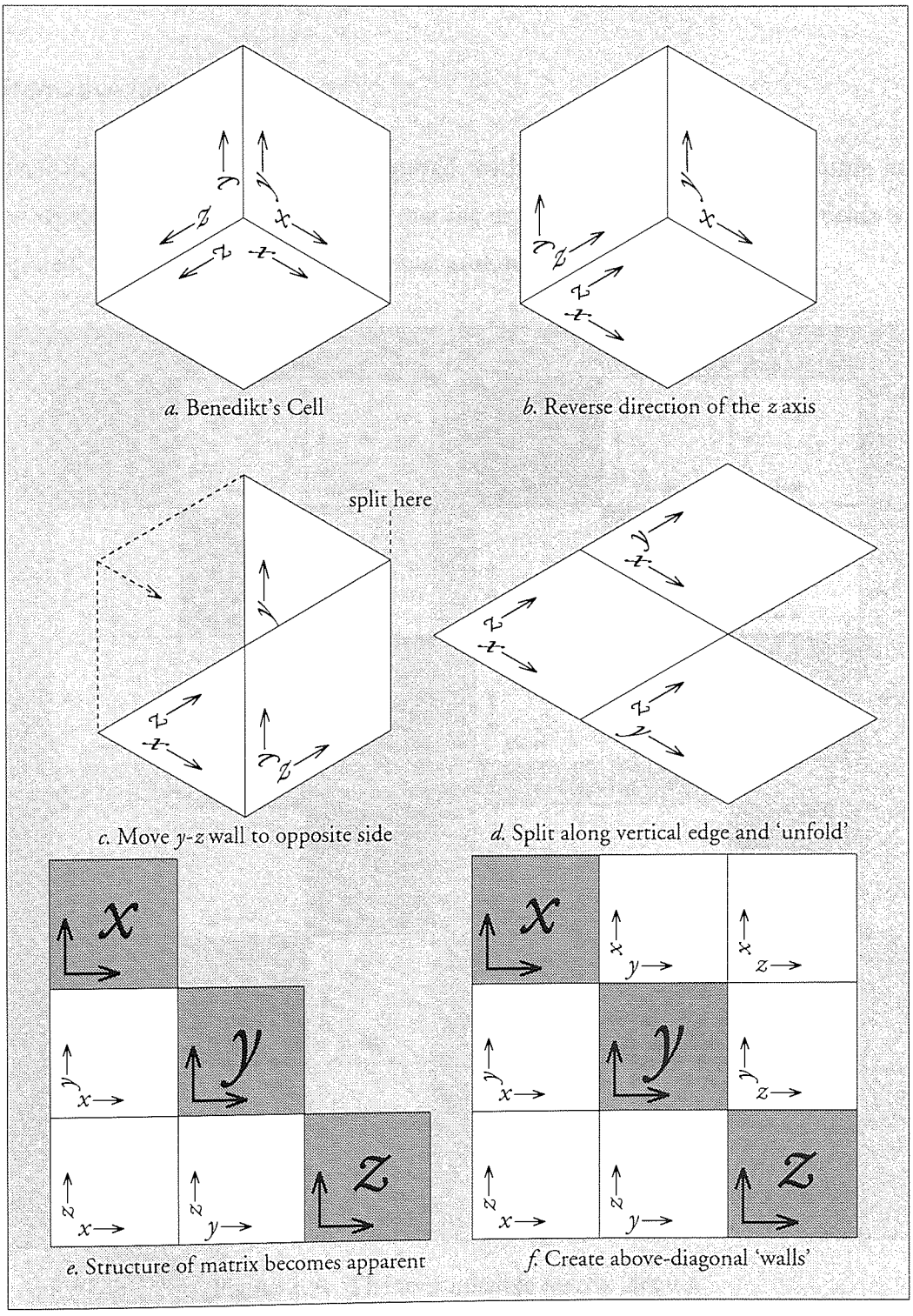


Figure 4.1 – Transformation from the 3-D Benediktine cell to a 2-D matrix

4.1.2 Extension to *n* dimensions

There is no reason why the matrix need have only three rows and columns – the structure clearly allows an arbitrary number of axes (two or more) to be displayed in the one matrix. By moving to 2-D, we are suddenly able to simultaneously visualise all possible pairwise projections of any number of data dimensions, albeit at a the cost of a reduced plot size.

4.1.3 Alternative layouts

Given that a matrix of walls was required, with the walls on the same column and row sharing the same *x* or *y* axis, and that the usual left-to-right ordering of fields was required, there were four rational layouts, as shown in figure 4.2.

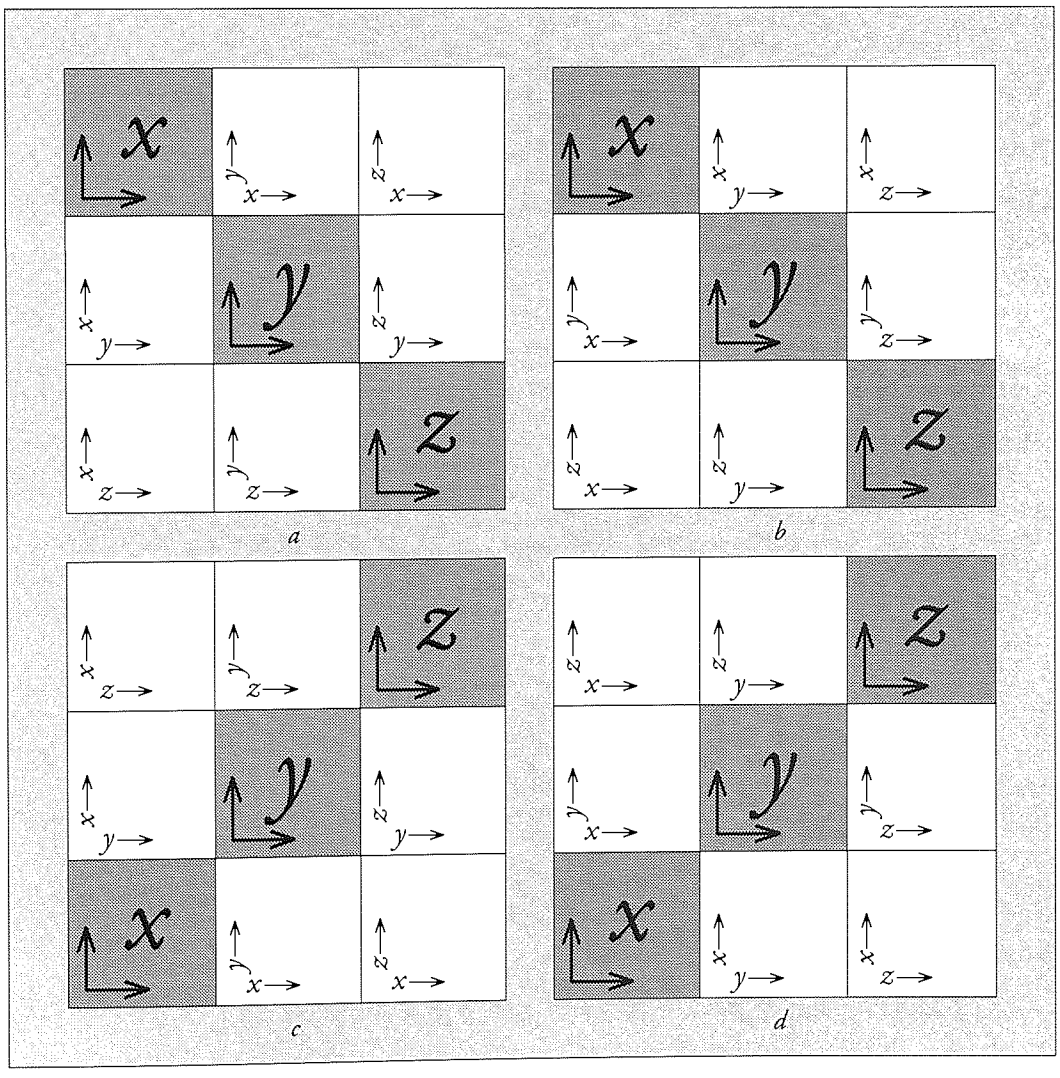


Figure 4.2 – The four possible matrix layouts

The walls of layouts *a* and *c* have their *x* axes determined by the row of the matrix in which they reside. This results in the axis running across the matrix, repeating itself at each wall. Though easier to understand than the axis layout discussed above and shown in layouts *b* and *d*, this would make it difficult to place labels on the axis. Layouts *c* and *d* were considered unsuitable as additional fields would be added at the top right, which is unintuitive. Layout *b* is the arrangement which was chosen, since it allows axis labelling and expansion is at the bottom right, as intuitively expected.

4.1.4 'Maden'

The visualisation system described in this chapter was given the name 'MADEN,' suggesting the involvement of 'matrices' and 'density'.

4.2 Implementation

4.2.1 Code structure

Although this system is conceptually based on the Benediktine cell program, none of the existing code was reused. The program was written using a fully-templated matrix library developed in the Aston Neural Computing Research Group by Mike Tipping and myself from original work by John van der Rest.

4.2.2 Overview display

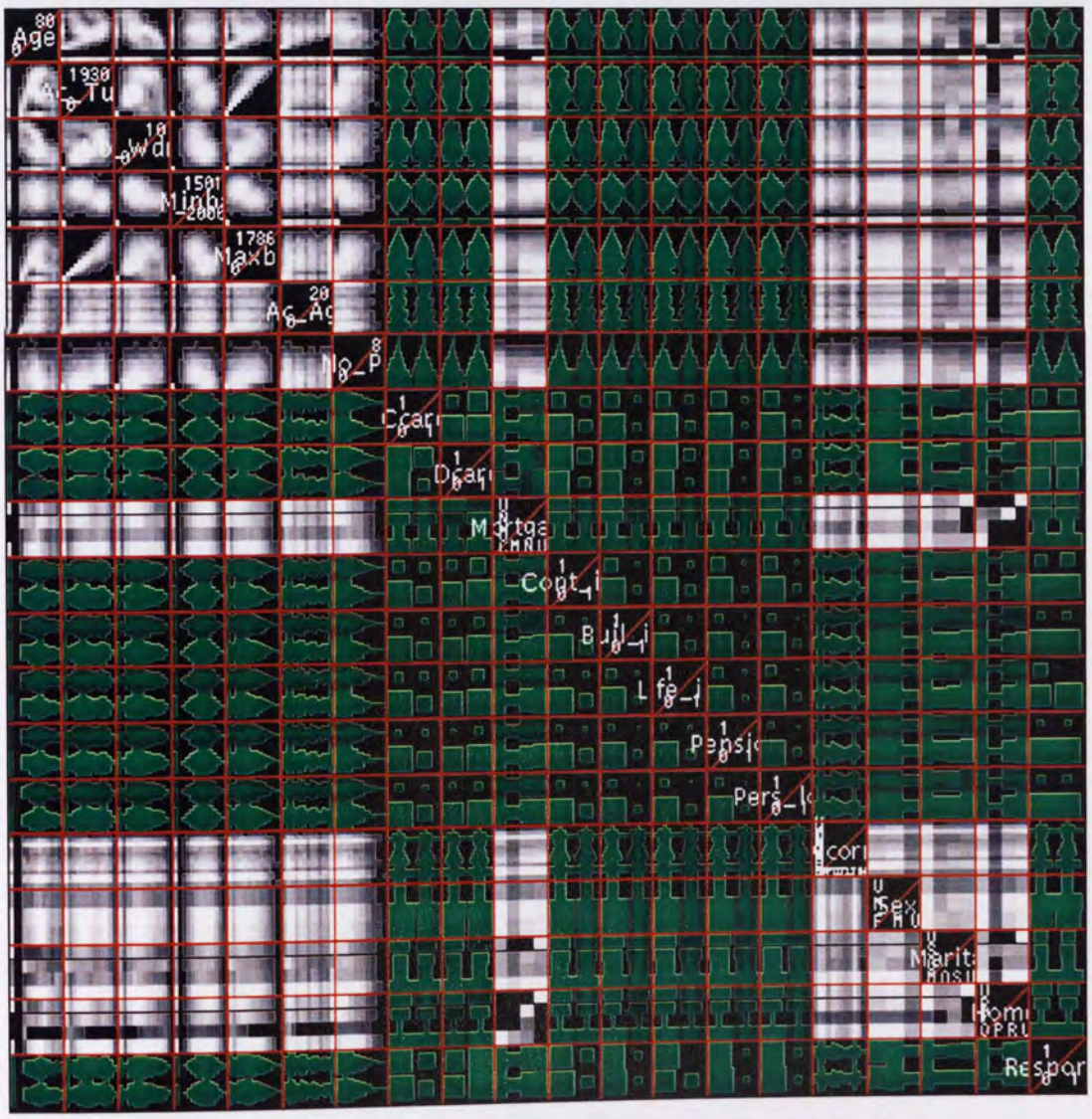


Plate 4.1 – Overview display showing the entire mail database

The main window of the MADEN system contains the matrix display described above, which is termed the *overview*. It is so named because it allows the user to obtain an

overview of the entire database under examination, since it is possible to display all the fields of the database at once, and hence (two versions of) *every possible projection* onto pairs of those fields. Plates 4.1 and 4.2 show two overview displays.

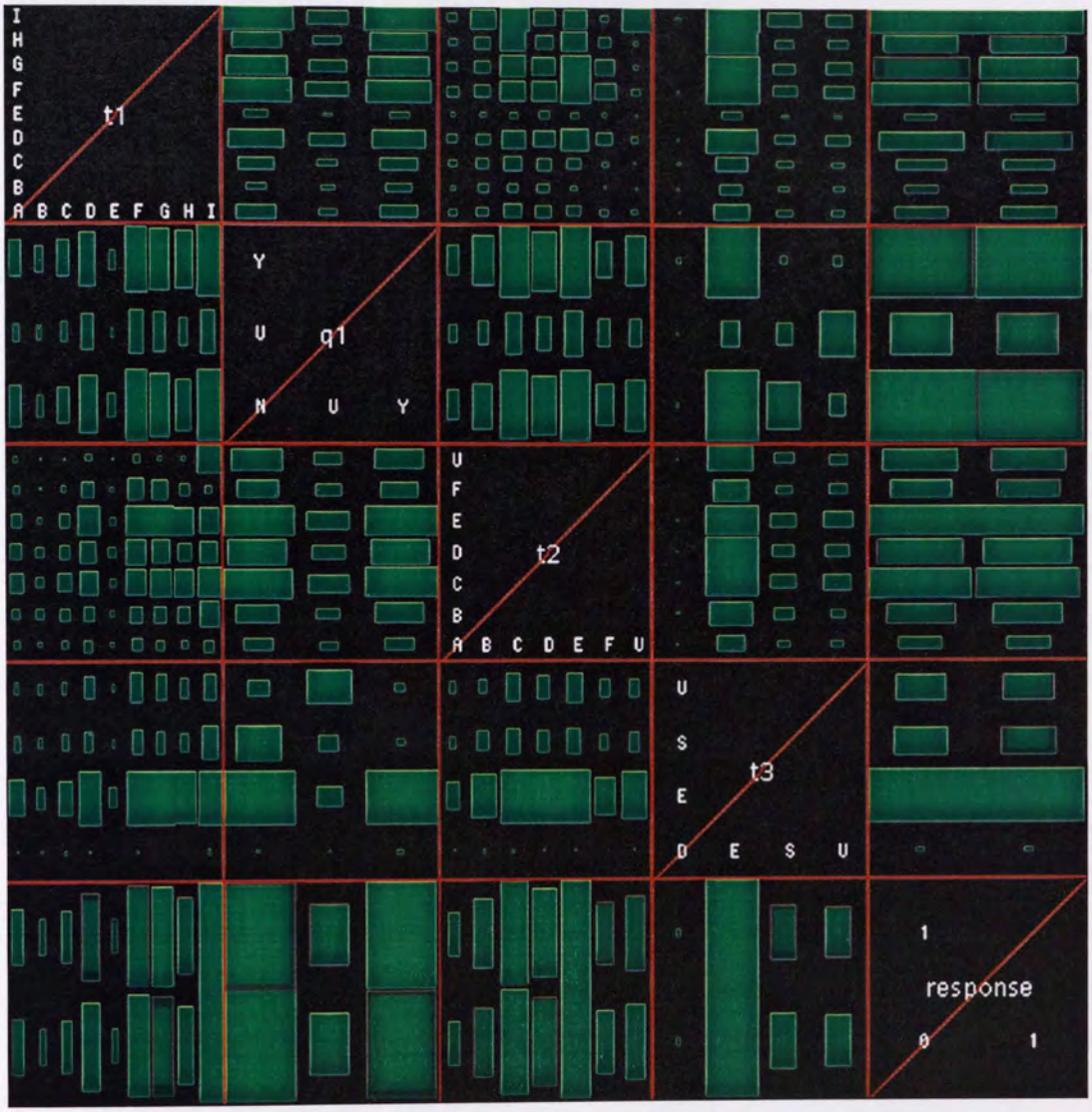


Plate 4.2 – Overview display showing five fields of the finance database

4.2.3 Density plots

Each off-diagonal element of the overview contains a *density plot*. These plots are based on the walls used in the cell visualiser, but with several enhancements over the implementation described in chapter 3, as detailed below.

4.2.3.1 Density matrix creation

Density plots are always displayed at the maximum resolution possible, and recalculated when the size of the display changes (either by the user resizing the overview window, or when the number of displayed fields changes, thereby changing the size of each row and column of the overview).

The density matrices are calculated as in the cell visualiser, with the fixed size of fifty bins changed to a variable size which allows two display pixels per density matrix row and column (two pixels was the smallest size for which the greyscale displays were clear). Thus if two continuous fields are used to generate a density matrix which will be displayed in a rectangle of 250×200 pixels (for there is no constraint that the overview window be square), the matrix generated will be 125×100 elements in size.

For integer and categorical fields, there is no change in the size of the density matrix, unless the size of the matrix would exceed the two-pixels size as calculated above. In this case, the field is treated as a continuous field and 'binned' to the two-pixel size. For example, an integer field with range 100 will result in a density matrix size of 100 as long as the displayed rectangle is at least 200 pixels wide. If the rectangle size is less than 200 pixels, the field will be binned.

The problem of skewed plots due to excessively large densities in a few locations in the density matrix, countered by the 'zero masking' option in the Benediktine visualiser, was tackled in a more flexible manner. The maximum density value in the matrix is identified, and the mean and standard deviation of all the other non-zero densities in the matrix are calculated. Then a threshold is set at two standard deviations greater than the mean, and all densities exceeding this threshold are clipped to the threshold (two standard deviations was chosen after experiments with real data). This process ensures that detail is not lost in the density plots, particularly if there is an extremely large density in one location, but the clipping is not restricted to 'hot spots' only occurring on the axes. This clipping is always enabled, removing one of the many options which could confuse a user of the cell visualiser.

4.2.3.2 Density plot display

The density plots are displayed in the same manner as the walls in the cell visualiser.

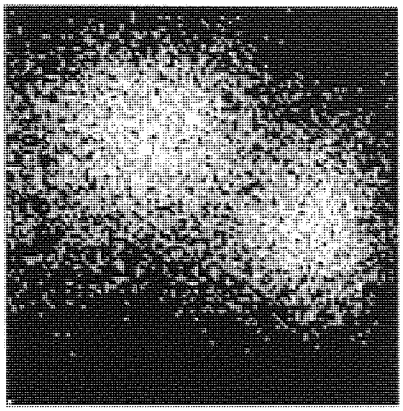


Figure 4.3 – Continuous/continuous density plot

- Continuous/continuous plots are shown as a seventeen-level greyscale (black, fifteen greys and white), with the grey level indexed by the square root of the density, as shown in figure 4.3. The square root operation is the 'enhancement' option of the previous system, which is taken out of user control and permanently enabled, since it was rarely disabled in actual use with real data.

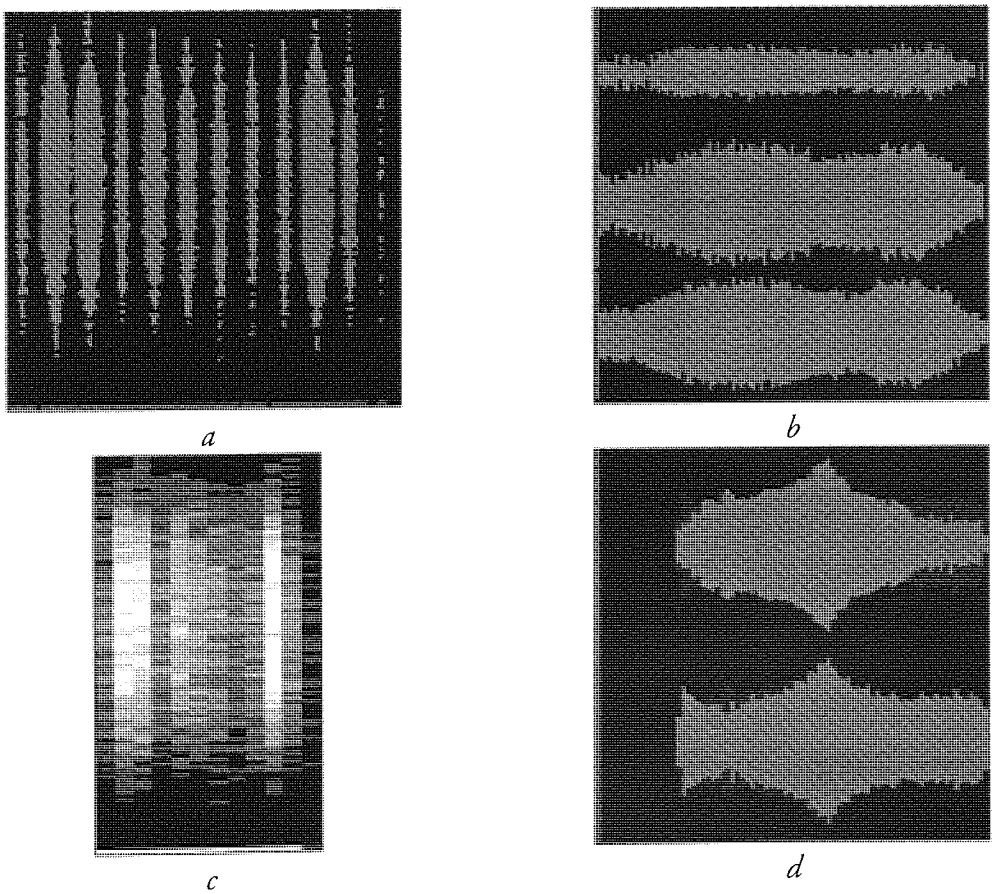


Figure 4.4 – Continuous/discrete density plots

- Continuous/discrete plots are shown as green strips of varying width, as shown in figure 4.4*a* and 4.4*b*. The width is again enhanced, since this gives a clearer representation of lower-density areas. Two modifications to the basic plot were made. Firstly, if the (maximum) width of each strip is less than ten pixels, the plot is shown as a series of full-width strips with greyscale shading to represent the density (figure 4.4*c*). This prevents small plots, or plots with many discrete values, becoming hard to interpret. Secondly, in the special case where the discrete field has two values, each strip is shown in dark green behind the other strip (figure 4.4*d*). This has the effect of highlighting the differences between the strips, and was introduced for this purpose, particularly when the discrete field is a binary response variable.

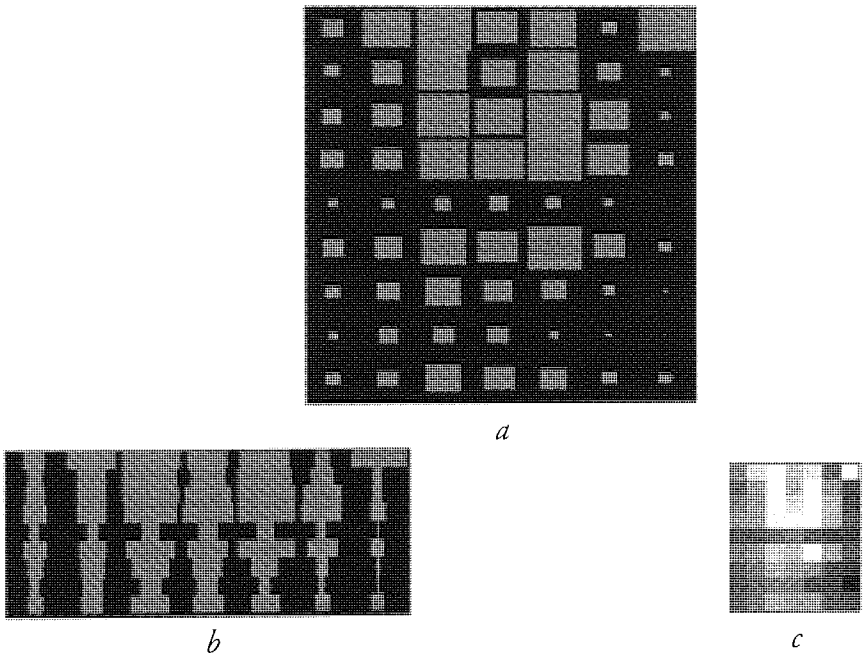


Figure 4.5 – Discrete/discrete density plots

- Discrete/discrete plots are shown as an array of rectangles whose linear sizes are proportional to the square root of the density at that location (thereby making the area of the rectangle proportional to the density), as shown in figure 4.5*a*. If the maximum rectangle size in either direction falls below ten pixels, the plot is shown as either a series of strips (if only one size is less than ten pixels) or as an array of full-sized greyscale rectangles (if both sizes are less than ten pixels), as shown in figure 4.5*b* and 4.5*c*.

4.2.4 Axis labelling

The major diagonal elements of the overview are used as *axis identifiers* for the row and column of the overview in which they are located, with the name of the field displayed in the centre of the identifier rectangle.

The MADEN system improves greatly on the cell visualiser by also providing permanent on-axis labelling of values. Continuous and integer fields are labelled with their minimum and maximum values – minimum values are displayed in the lower left of the identifier rectangle, maxima in the upper right, as shown in figure 4.6*a*. These locations apply in both orientations of the axis (as the *y* axis for the density plots in the same row as the identifier, and the *x* axis for plots in the same column). For integer fields, if feasible, the labels are centred on the position of the value on the axis.

a *b*

Figure 4.6 – Examples of axis identifiers

For categorical fields, the category labels are shown along the left and bottom edges of the identifier, centred on the position of the category on the axes, as shown in figures 4.6*b* and 4.2. This choice of edges allows the first category label to apply to both orientations, as well being the intuitive labelling for plots with their origins in the lower left corner.

4.2.5 Axis selection and ordering

Initially, the overview window was made to show every field when it was created, as seen in figure 4.1. A versatile interface was created to allow the fields to be reorganised into any desired sequence: pressing the right mouse button in an axis identifier brings up a popup menu similar to the one shown in figure 4.7. The 'Delete' option removes the field from the overview, the 'Move' submenu allows the field to be repositioned anywhere in the overview field order, and a button at the top of the overview window restores the initial selection of all fields in their normal order.

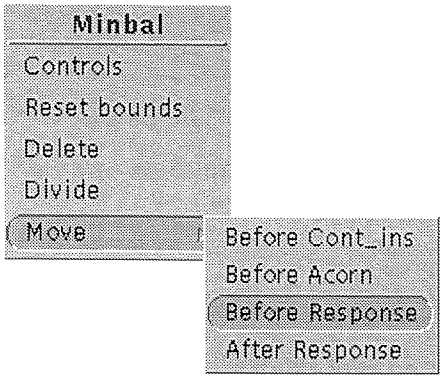


Figure 4.7 – An axis popup menu

In use, however, the display of all possible fields was found to have two major drawbacks: it took a long time to calculate and display, and it resulted in very small density plots. Inevitably, the user began by deleting most of the fields from the overview. Therefore the initial conditions were modified so that only five fields are shown when the program starts. Four fields are chosen using techniques described in the next section, and the fifth displayed field is the last field, i.e. the response field. This results in an initial display as seen in plate 4.2.

A method was now required to add individual fields to the overview. A solution was developed, in the form of a 'Displayed axis window', as shown in figure 4.8. The window contains a toggle button for each field in the database, which the user can press to choose which fields are required in the overview. Once a choice has been made, the 'Update' button can be pressed to change the overview. The 'Reset' button restores the initial five fields to the overview. This axis selection method proved to be easy and convenient to use, and since no better method was found, it was left in the system.

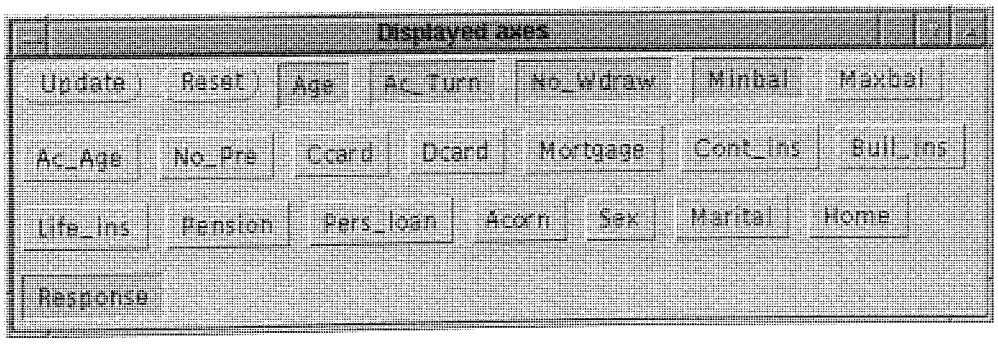


Figure 4.8 – Displayed axes window for the mail database

4.2.6 Initial field choice

The choice of which fields to display in the initial overview is made on the basis of the deviation of their distributions from uniform and normal distributions which are defined as 'uninteresting'.

Firstly, a vector containing the distribution of the field in question is created. For discrete fields of less than 100 values, the distribution is binned into the same number of bins as there are values; for discrete fields with more than 100 values, and continuous fields, the distribution is binned into 100 bins. The distribution vector is then normalised to sum to unity. In the following equations, x_i is the ith element of this normalised vector, n_b is the number of bins, and $\bar{x}$ is the mean of the vector elements, which is equal to $\frac{1}{n_b}$ due to the normalisation.

Two indices are calculated. $q_{nonuniform}$ is defined in equation 4.1 and is a measure of the deviation of x from a uniform distribution; $q_{nonnormal}$ is defined in equation 4.2 and is a measure of the deviation of x from a normal distribution with the same mean and variance.

$$q_{nonuniform} = \sum_{i=1}^{n_b} \left| x_i - \bar{x} \right| \tag{4.1}$$

$$q_{nonnormal} = \sum_{i=1}^{n_b} \left| x_i - \frac{\int_{x=i-\frac{1}{2}}^{i+\frac{1}{2}} \mathbb{N}_{\mu,\sigma}(x)}{\int_{x=\frac{1}{2}}^{n_b+\frac{1}{2}} \mathbb{N}_{\mu,\sigma}(x)} \right| \tag{4.2}$$

where $\mu = \sum_{i=1}^{n_b} i x_i \qquad \sigma^2 = \sum_{i=1}^{n_b} i^2 x_i - \mu^2$

A combined index, $q_{uninteresting}$, is then calculated as in equation 4.3, which gives a measure from zero to unity, where the most 'interesting' fields give the lowest values.

$$q_{uninteresting} = \frac{1}{1 + q_{nonuniform}} + \frac{1}{1 + q_{nonnormal}} - 1 \tag{4.3}$$

The four fields with the lowest values of $q_{uninteresting}$ can then be selected for display in the initial overview.

4.2.7 Overlays

Overlays are chosen from a menu at the top of the overview window, as demonstrated in figure 4.9. As shown, this menu includes a 'none' option to disable the overlay.

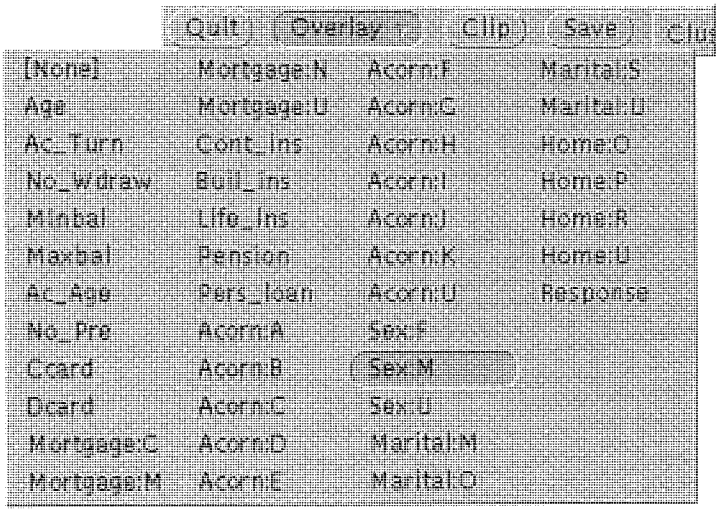


Figure 4.9 – Overview overlay menu

The overlay is immediately applied to all density plots in the overview. User control over which plots are overlaid was not provided as it would be far too complex for a matrix of hundreds of plots, such as plate 4.1, which shows 380 plots.

The overlay is shown in the same manner as in the cell visualiser, using a blue-red colour scale. In the MADEN system, there are seventeen levels on the colour scale (to correspond with the seventeen levels of grey in the greyscale). For plots which would be shown in greyscale (rather than in green), four levels of brightness are provided in the colour scale. Plots with a discrete field which has only two categories retain the dark green background strip described in section 4.2.3.2. An overview with an overlay is shown in plate 4.3, overleaf.

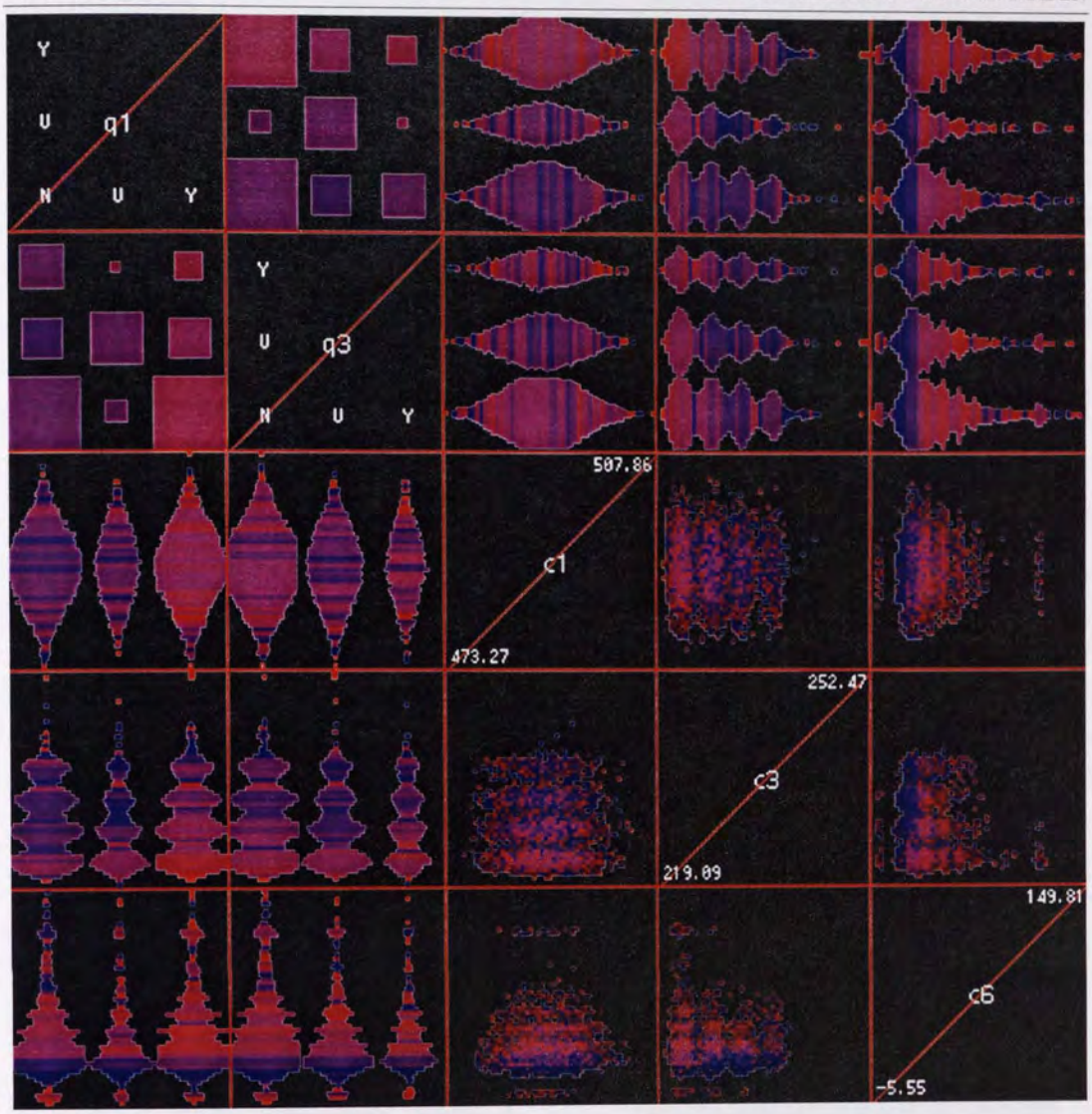


Plate 4.3 – Example of an overview with an overlay

With a binary overlay field (i.e. one with two categories), using red for one category and blue for the other is acceptable. However, in the cell visualiser, when using an overlay field with three categories, the implicit ordering of the categories (due to the internal representation as integers) could result in a mid-scale colouring which might represent a mixture of categories one and three, or just category two. This mixing is almost certainly not valid (e.g. half female and half unknown is not equivalent to all male), and so the MADEN overlay generation was modified. For categorical fields with more than two categories, the user must choose which category to use to generate the overlay, as shown in figure 4.9. This results in the chosen category being shown in red, and all the others in blue.

When the system was used to visualise the RAE database, an alternative colour scale was added, as described in section 4.3.3.2.

4.2.8 Enlargements

In the cell visualiser, the user can manoeuvre the vehicle towards a particular wall in order to get a closer view. In the MADEN system, it is not possible to magnify a density plot in the overview, except by physically enlarging the window or by reducing the number of fields shown in the overview. Instead, the user can create *enlargements* of density plots, simply by clicking the right mouse button on the plot to be enlarged.

Each enlargement is shown in its own window, which contains a density plot identical (except in size and resolution) to the plot in the overview whence it was generated. Below and to the left of the plot are labels indicating the field assigned to each axis and the values of that axis. Categorical fields are labelled with the field categories, integer fields with their minimum and maximum values, and continuous fields by nine labels distributed along the axis. Two enlargements are shown in figure 4.10.

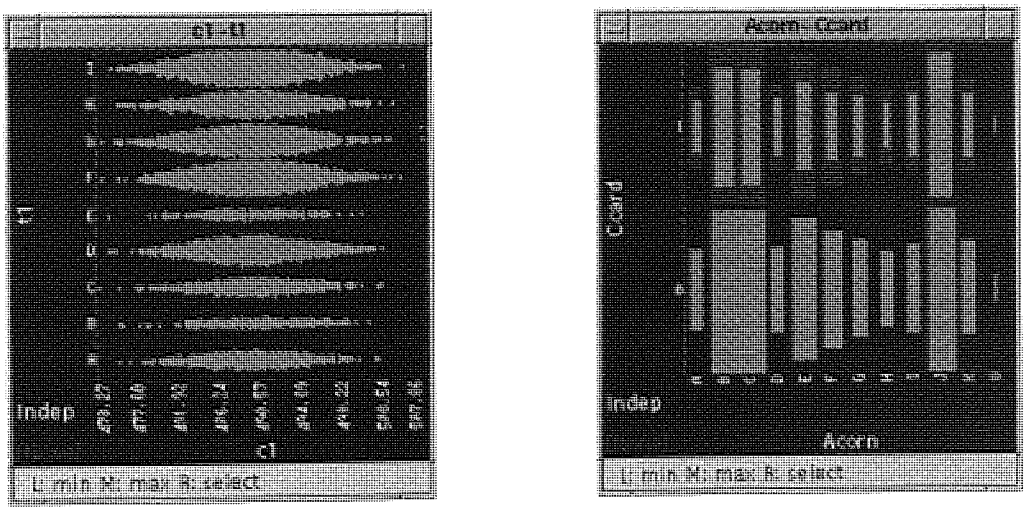


Figure 4.10 – Example enlargements

The overlay (or lack of overlay) in the overview when the enlargement was created is preserved in the enlargement density plot. The field used for the overlay is shown in the lower left corner of the window. If the overview overlay subsequently changes, the enlargement's overlay does not, thereby allowing overlays to be compared in a manner impossible in the cell visualiser. Indeed, multiple enlargements of the same density plot, each with a different overlay, may be opened, as shown in plate 4.4.

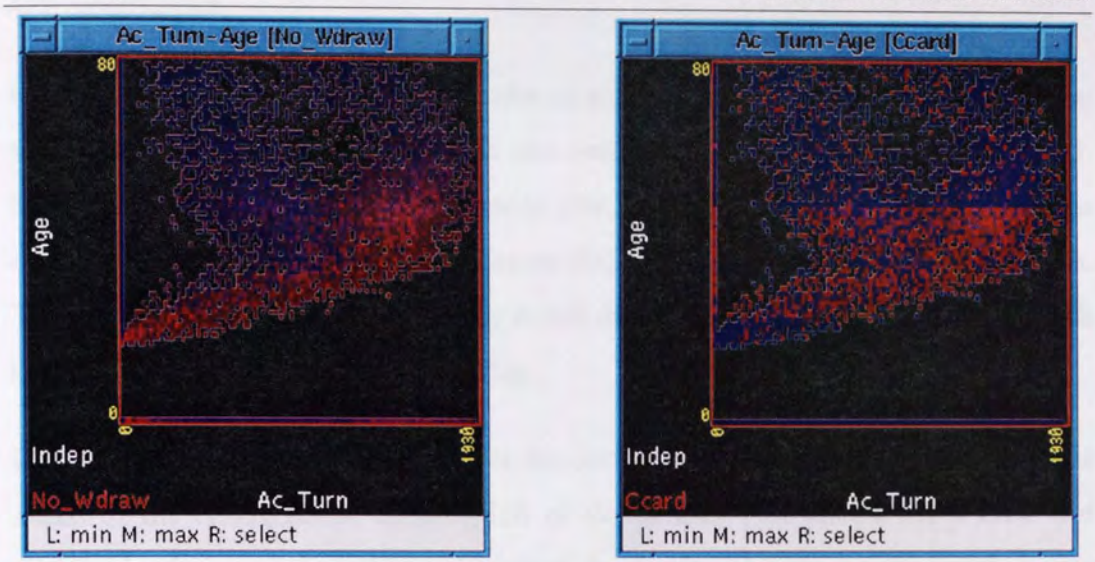


Plate 4.4 – Two enlargements of the same density plot with different overlays

The user can resize enlargement windows to gain a plot of arbitrary resolution, as demonstrated in figure 4.11, which evidently bears more resemblance to a traditional scatter plot than the small density plots.

Figure 4.11 – A highly enlarged density plot

4.2.8.1 Axis modification

In testing, it proved tedious that in order to generate a desired enlargement, the two axis fields had to be made visible on the overview, and the overlay field selected, before clicking on the appropriate density plot. The decision was thus taken to allow the axis fields and the overlay field to be modified from within enlargement windows. The initial choice of fields and overlay is still determined by the location of the click in the overview and the overview overlay.

Clicking the right mouse button below the density plot pops up a menu to select the x axis of the enlargement, clicking left of the density plot selects the y axis, and clicking in the area below and to the left of the density plot (in the lower left corner of the window) allows the overlay to be changed or disabled.

In this way, the user need only create as many enlargement windows as are required on-screen at once (maybe two or three), and can change the plot shown in each window as necessary.

4.2.9 Selection

In the cell visualiser, the probe is used to make selections along the three displayed axes. In the MADEN system, a new method was required, due to the large and variable number of axes which can be displayed.

The cell probe selects a range of only the three displayed fields, and is initially one unit wide along each. In MADEN, each axis (whether displayed or not) has an associated selection range. Initially this range covers the entire range of the field, since to start otherwise would require the user to increase the size of the selection on virtually all axes before attempting to make a valid selection.

4.2.9.1 Selection display

The selection is outlined in yellow, like the probe in the previous system. However, it is shown in different ways in different areas of the display.

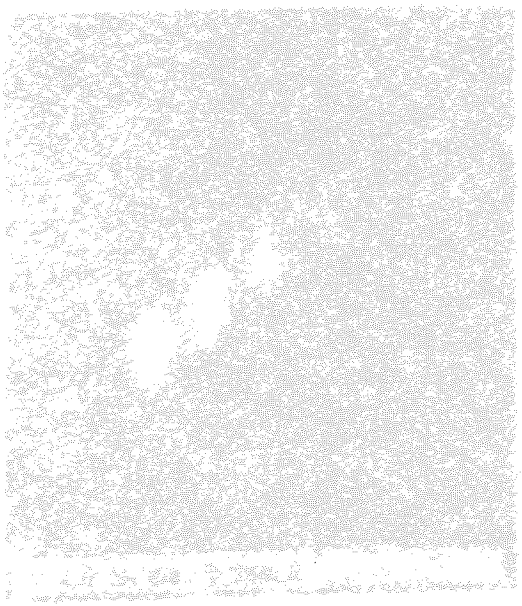
Each density plot in the overview shows a yellow rectangle which encloses the selection along the two axes of the plot. For discrete fields, the edges of the rectangle are aligned between the discrete values, rather than on them, in order to clearly show

which values are enclosed by the selection. If both the top and bottom edges, or both the left and right edges, of the rectangle lie at the extremes of the plot (i.e. the selection covers the whole range of the field), that pair of edges is not shown. In this way, the initial, total, selection does not show at all, and any changes in the selection which are made are immediately apparent.

The axis identifiers in the overview cannot use the rectangle display method, since they only show one field. Instead, two pairs of yellow lines are drawn across the identifier rectangle, at the horizontal and vertical extremes of the selection. These lines cross on the lower left to upper right diagonal of the rectangle, and in order to emphasise that this diagonal is effectively the axis line, it is drawn in red.

Plate 4.5 illustrates the display of selections. Three axes have selections made: `t3` has the 'S' category selected, `c2` has the range of (approximately) 0–13 selected, and `c3` has a selection made from its minimum up to about 232. Various density plots in this overview demonstrate all the selection display techniques discussed above, including one- and two-axis selections, some of which extend to one extreme of the range, selections of continuous and discrete fields, and selection indication on axis identifiers.

In enlargements, the selection boundaries are drawn right across the density plot, to emphasise the location where they meet the axes, as shown in plate 4.6.



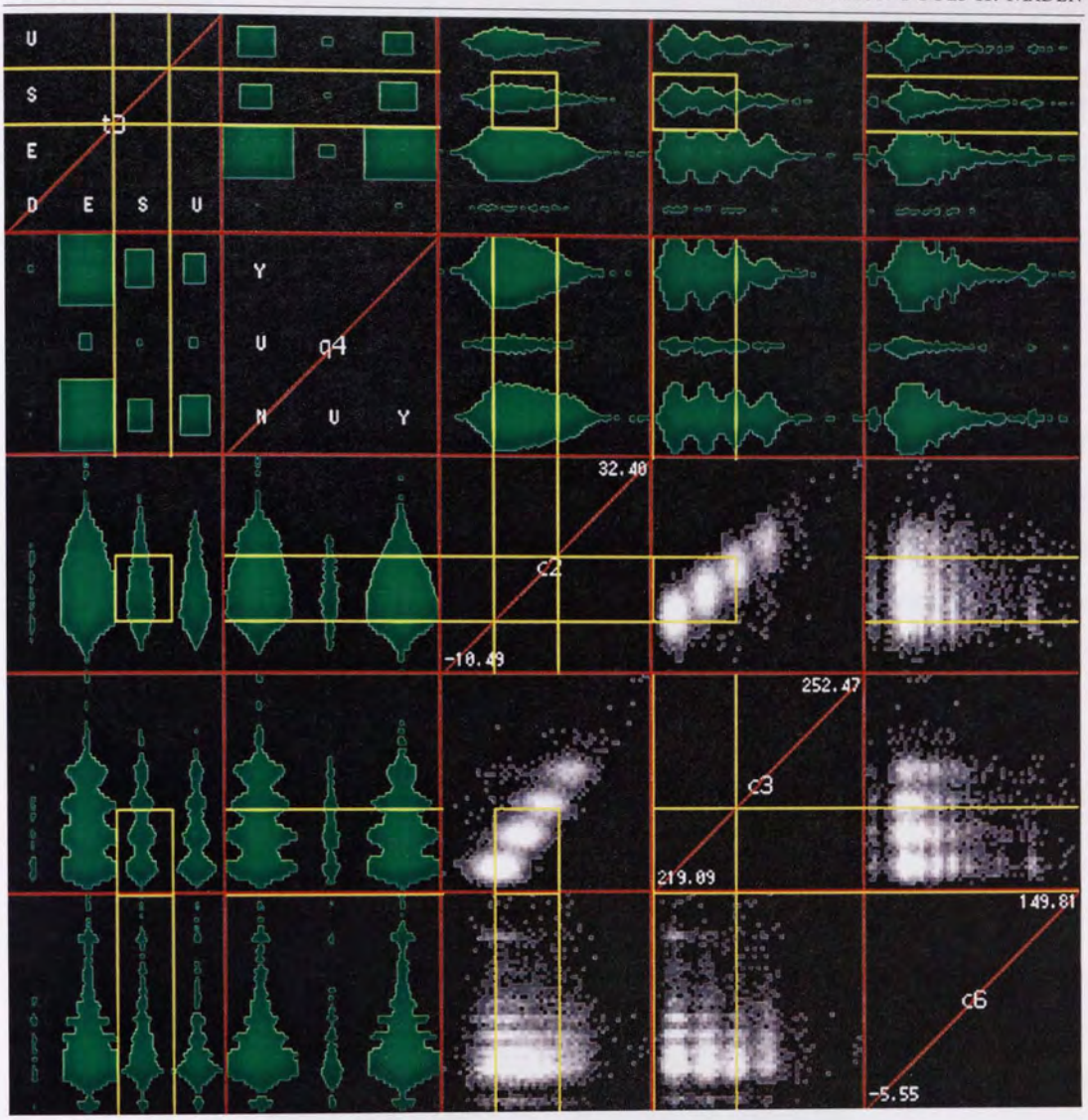


Plate 4.5 – Overview showing a selection made on three axes

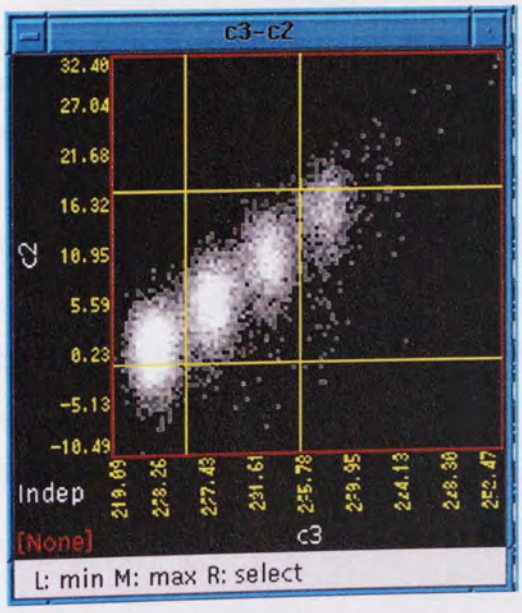


Plate 4.6 – Enlargement window showing a selection

4.2.9.2 Selection control

Selections are made using the left and middle mouse buttons: the left mouse button sets the lower bound of the selection, the middle button the upper bound.

Clicking in an axis identifier sets one bound of the selection of the field shown on that axis. The value is calculated by projecting the click location onto the red diagonal axis line. In the case of discrete fields, the closest discrete boundary is chosen.

Clicking in a density plot, whether on the overview or in an enlargement, sets the bound of *two* fields, one for each axis of the plot in which the click occurs. Thus a 2-D selection can be specified with two clicks in the same density plot – the left mouse button at the lower left corner of the selection, and the middle mouse button at the upper right corner.

An alternative method for selection is the field control window, which is made visible through the axis popup menu (as seen in figure 4.7). An example of this window is shown in figure 4.12.

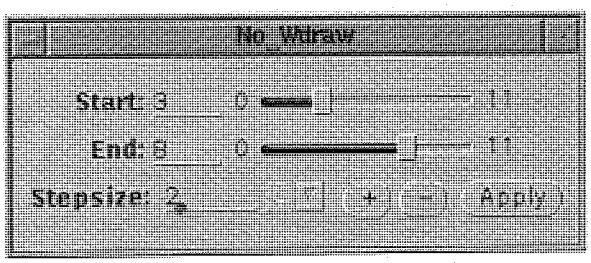


Figure 4.12 – A field control window

The field control window displays the bounds of the current selection. The user can change these bounds directly by typing into the numeric data fields or by moving the slider controls. Fine control over the movement of the selection is provided by the 'step size' data field and the plus and minus buttons. Pressing one of these buttons changes both the upper and lower bounds of the selection by the step size, either upwards or downwards. This mechanism allows, for example, a selection initially set to 20–30 to be taken through the sequence 20–30, 25–35, 30–40 or 20–30, 30–40, 40–50 or 20–30, 35–45, 50–60 by setting the step size to 5, 10 or 15 respectively.

Due to limitations of *XView*, the field control windows always use integers to indicate the selection bounds. This has two consequences: categorical fields are referred to by ordinal number (but since the selected categories are clearly indicated in the axis indicator, this is not a problem), and continuous fields are rounded to the nearest

integer, which does not has a major effect on usability (non-integer selection bounds are possible via mouse clicks on density plots or axis identifiers).

Whenever the selection is changed, either by clicking or through a field control window, the yellow selection indicators in the overview and enlargements immediately change to reflect the selection, and the numeric values in the field control windows immediately update to show the new bounds (rounded if appropriate). As will be seen in section 4.2.14, the number of records currently selected is displayed at all times.

4.2.10 Dependent enlargements

The dependent wall was an important feature of the cell visualiser, which had to be carried over in some form to the MADEN system. Rather than attempt to update the entire overview display when the selection changed, it was decided to allow only enlargements to be dependent on the probe selection, in much the same way as individual walls in the cell visualiser. A dependent enlargement's density plot displays the projected density of only those records currently selected.

The lower left corner of an enlargement window displays either 'Indep' or 'DEP' to indicate whether its density plot is dependent upon the current selection or not. Clicking the left mouse button on this text toggles the state and updates the display immediately. Whenever the selection changes, for whatever reason, all dependent enlargements are updated. The use of dependent enlargements will be demonstrated in section 4.3.1.3.

4.2.11 Clipping

Clipping is the MADEN equivalent of opening a subspace in the cell visualiser. When the user presses the 'clip' button at the top of the overview window, a new overview is created, containing only those records which are enclosed in the current selection.

Additionally, the clip operation may be used to eliminate fields from the database. If the shift key is held down while the clip button is pressed, the new overview will again contain only those records in the selection, but in this case only those fields currently visible in the first overview will exist in the new overview.

4.2.12 Data saving

Each overview window also has a 'save' button, which outputs the current database to a file. This allows the user to clip the database (both its records and fields, if required) and maybe perform some transformations on the data (as detailed in later chapters), and then to save the results for further analysis by MADEN or otherwise.

4.2.13 Window deletion

Each overview window has a 'quit' button which deletes the window and any sub-windows belonging to it (windows which share the same database as the overview – i.e. enlargements and other windows discussed in later chapters). However, each overview is completely independent of all other overviews displayed by the system: unlike the cell visualiser, there is no hierarchy of overviews, and in particular the first overview is not treated in any special way. Deleting one overview has no effect upon any others, and the system continues to run until there are no overviews left.

4.2.14 Footers

To simplify use of the system, the left hand footer of each window (overviews, enlargements and other windows to be discussed in later chapters) displays a key of what effect each mouse button will have in that window, as demonstrated in figure 4.13, which shows the footer of the overview window.

Figure 4.13 – Left hand footer of the overview window

The right hand footer of the overview window (shown in figure 4.14) displays statistics about the overview:

- the number of records in the database, i.e. the number shown in the overview
- the number of selected records (initially all the records in the database)
- the number of 'highlighted' records (see section 4.3.3.5, later)
- the mean value of the response field across all the records in the database
- the mean value of the response field for the records currently selected

These figures are continually updated, allowing the user to immediately see the effect of making a selection, particularly when attempting to select areas of high response in a database.

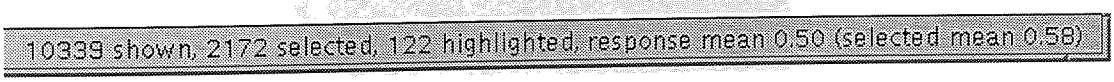


Figure 4.14 – Right hand footer of the overview window

4.3 Use with Real Data

4.3.1 Mail database

Exploration of databases with MADEN was begun by examining an overview showing all the fields of the database. For the mail database, this is the display seen in plate 4.1 on page 95. This shows the entire database, something which was completely impossible with the cell visualiser.

4.3.1.1 Use of enlargements

Some interesting plots which were seen in the overview were enlarged and are discussed below.

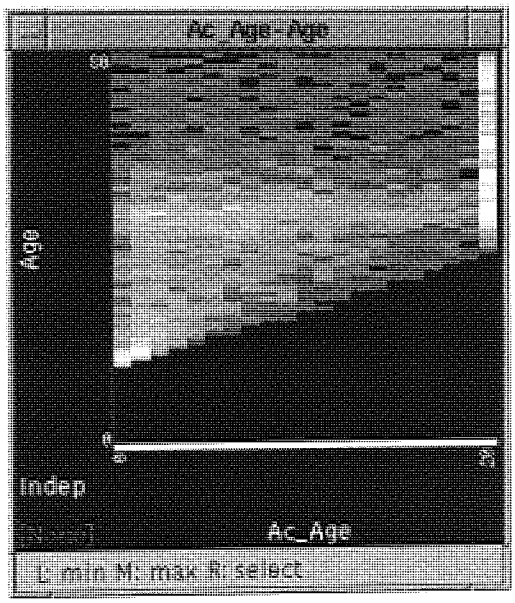


Figure 4.15 – Ac_Age–Age enlargement

Figure 4.15 demonstrates that the data that is present in the database has some consistency: no account age is less than seventeen years older than the customer's age. Clearly, customers have to be at least seventeen before they can open accounts with this institution. The plot also shows that there are a large number of accounts aged twenty years. This may indicate that the account in question was first offered twenty years ago, or (more likely) that this represents 'twenty or more' years.

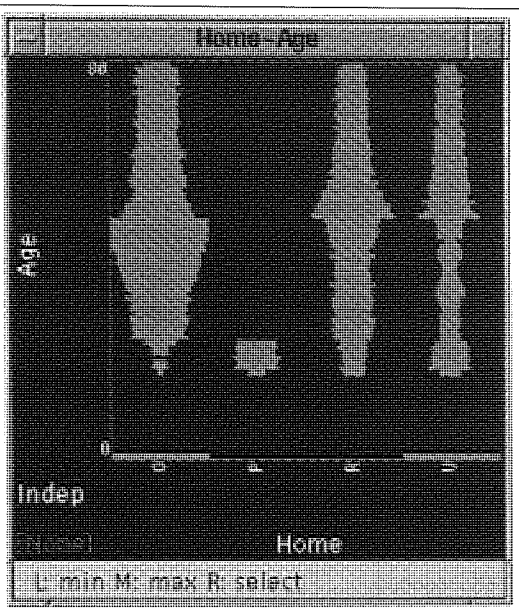


Figure 4.16 – Home–Age enlargement

Figure 4.16 enables us to surmise the meaning of the P category of the Home field. O and R represent home owners and those who rent their home, and U is used where the home status is unknown. P only occurs with customers aged under 24. It therefore appears this category is almost certainly customers who live with their parents.

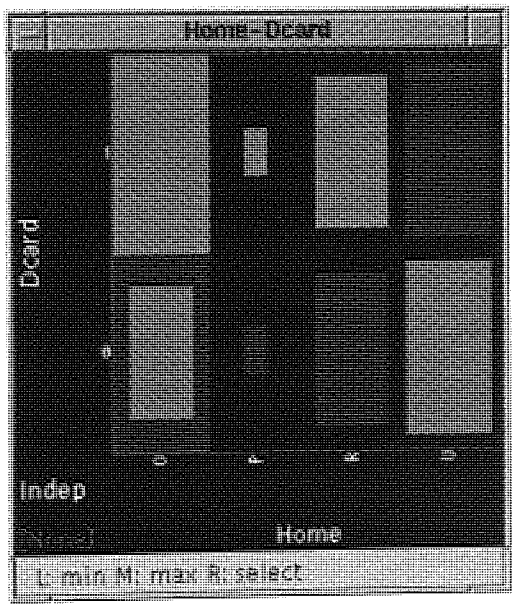


Figure 4.17 – Home–Dcard enlargement

The plot in figure 4.17 is very revealing. Some people who own their homes have debit cards, some do not. However, *everyone* who lives with their parents or rents their house has a debit card. This is the sort of information that a marketing manager might find very useful. Those customers whose home status is unknown are recorded as not having a debit card. This is probably because neither piece of information is known, and 'no debit card' is the default.

4.3.1.2 Use of overlays

By overlaying the response field on the overview shown in plate 4.1, it was difficult to identify many density plots which showed an interesting distribution of responders. One plot which does reveal some features is shown in plate 4.7 overleaf, which shows the plot of No_Wdraw against Ac_Turn with Response overlaid.

Two roughly elliptical clusters of responders can be seen, one with high turnover, the other with lower turnover. Plotting against the number of ATM withdrawals creates the elliptical grouping through the interrelationship between the two axis variables.

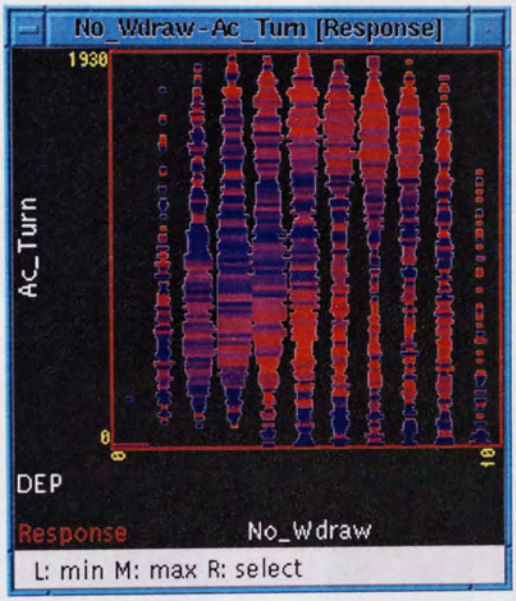


Plate 4.7 – No_Wdraw–Ac_Turn enlargement with Response overlaid

4.3.1.3 Use of dependent enlargements

After some exploration of likely dependent enlargements, the plot of Home against Marital with Response overlaid was chosen as an illustrative example. A sequence of four plots dependent on a changing Age selection are shown in plate 4.8.

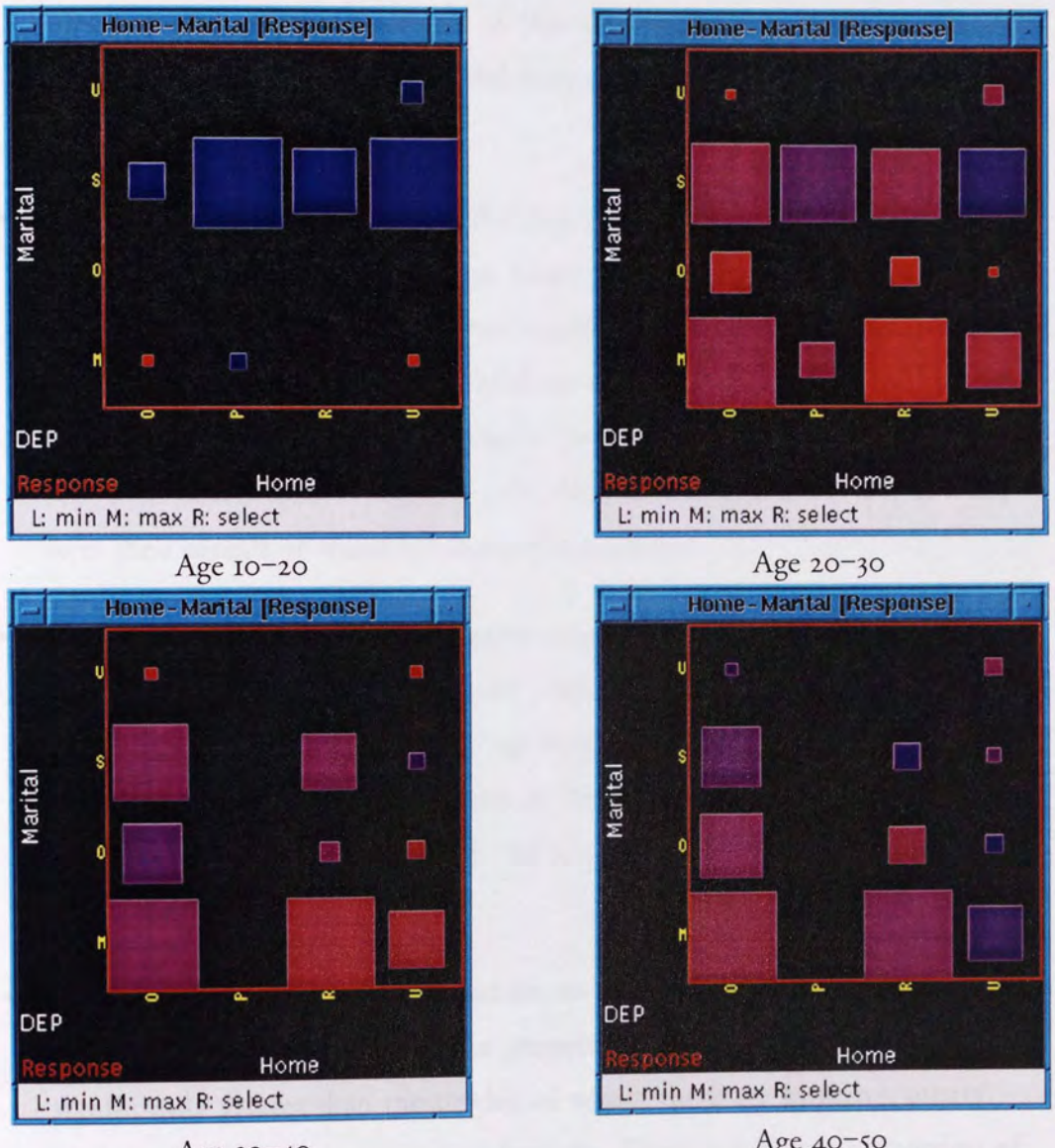


Plate 4.8 – Home–Marital enlargement with Response overlay, dependent on selection made on Age

Many observations can be drawn from the dependent plots in plate 4.8:

- For the 185 customers aged 10–20 (actually 17–20 since there are no customers under 17), the vast majority are single. The largest proportion of those whose home status is known live with their parents, though many rent and some own their homes. A small number are married, but none of these rent. Virtually none of the customers in this age bracket responded to the life insurance mail shot – those that did were generally married and presumably thinking about their future.

- For age 20–30, many more of the 869 customers are married, and they generally own or rent their homes. Many are still single, however, with an even spread of home types. The 'other' marital category now has some customers in it – probably divorcees. None of them admitted to returning to live with their parents, though. As for response to the mail shot, non-single people who rent their homes seem to respond more than others, particularly singles living with their parents or whose home status is unknown.

- In the 30–40 age range, which contains 1224 customers, there are no customers living with their parents (as noted previously), but apart from that the distribution is similar to the 20–30 age range. The response to the mail shot is not clear-cut, though home-owners in the 'other' marital category are less likely to respond: paying alimony and a mortgage leaves little funds for life insurance.

- The plot of the 2539 40–50s is similar to those for all ages over 40, so the other decades are not shown. The proportion of married customers is now significantly greater than the singles, of whom there are as many 'others' – likely to be both divorcees and widow(er)s. There is a higher proportion of 'others' in rented accommodation – maybe the house had to be sold to pay the alimony. Response is generally higher than for younger customers, particularly non-singles. Singles who are renting have a low response.

The above observations could be used to the company's advantage, by allowing a marketing manager to identify the patterns of customers in the database, and to see which groups of them are likely to respond to the mail shot.

4.3.2 Finance database

Figure 4.18 shows the overview of the entire finance database. Many interesting features can be seen, but the interpretation and understanding of them is severely limited by the lack of information regarding the meaning of the fields.

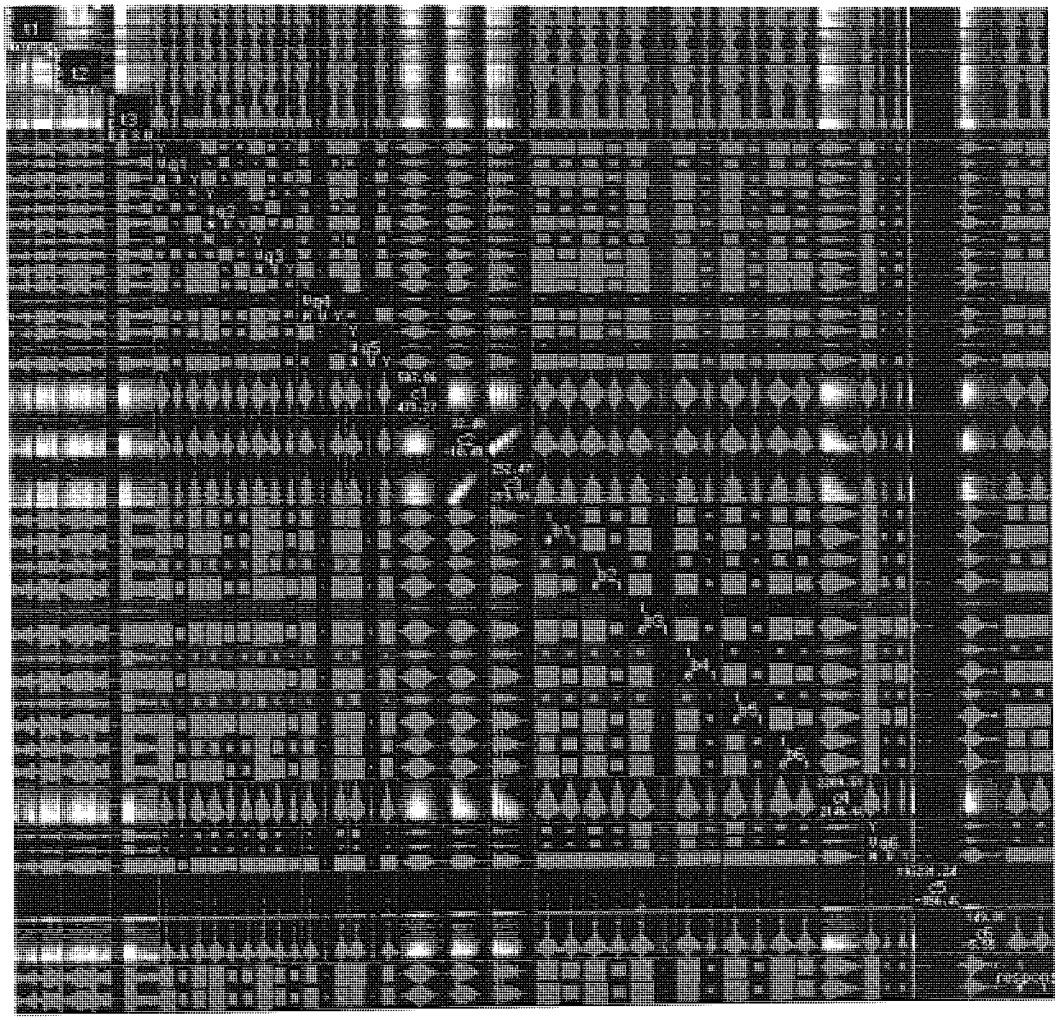
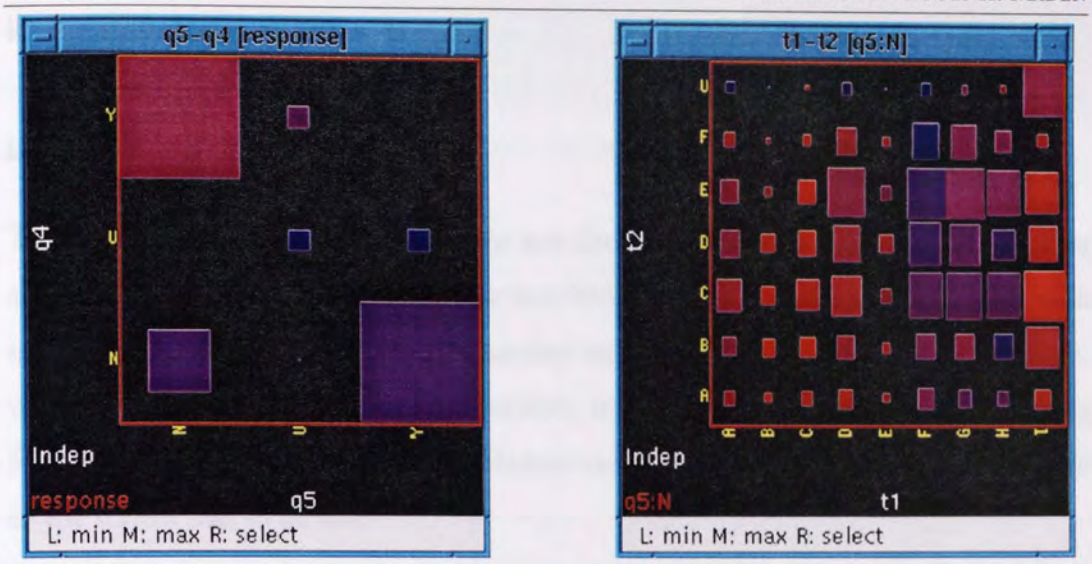


Figure 4.18 – Overview of the entire finance database



a *b*

Plate 4.9 – Two enlargements from the finance database

Plate 4.9 shows two enlargements which might be of interest to a user of the system who knew what the database fields meant.

The enlargement in plate 4.9*a* shows that no customers answer 'yes' to both questions q4 and q5, but a few answer 'no' to both. The highest response rate is from those customers who answered 'yes' to q4 and 'no' to q5, the lowest is from those whose answer to q4 is unknown.

The enlargement in plate 4.9*b* shows the distribution of t1 against t2, but instead of the response, it uses as its overlay the 'no' response to q5. A clear pattern of customers who answered 'no' to this question can be seen, and no doubt would be of use to a marketing manager investigating the database.

Further investigation of the finance database was certainly possible, but ultimately worthless, due to the lack of meaningful interpretation. This database will be returned to in later chapters.

4.3.3 RAE database

4.3.3.1 Difficulties

The major problem with visualising the RAE database is the number of fields. Showing all 84 on one overview is possible, but very little useful information can be seen with so many tiny density plots. Dimensionality reduction, the solution to this problem, will be covered in chapters 6 and 7. However, it is possible to discover some interesting features of the database using the raw MADEN viewer, though a number of modifications to the system had to be made.

4.3.3.2 Alternate overlay colour scale

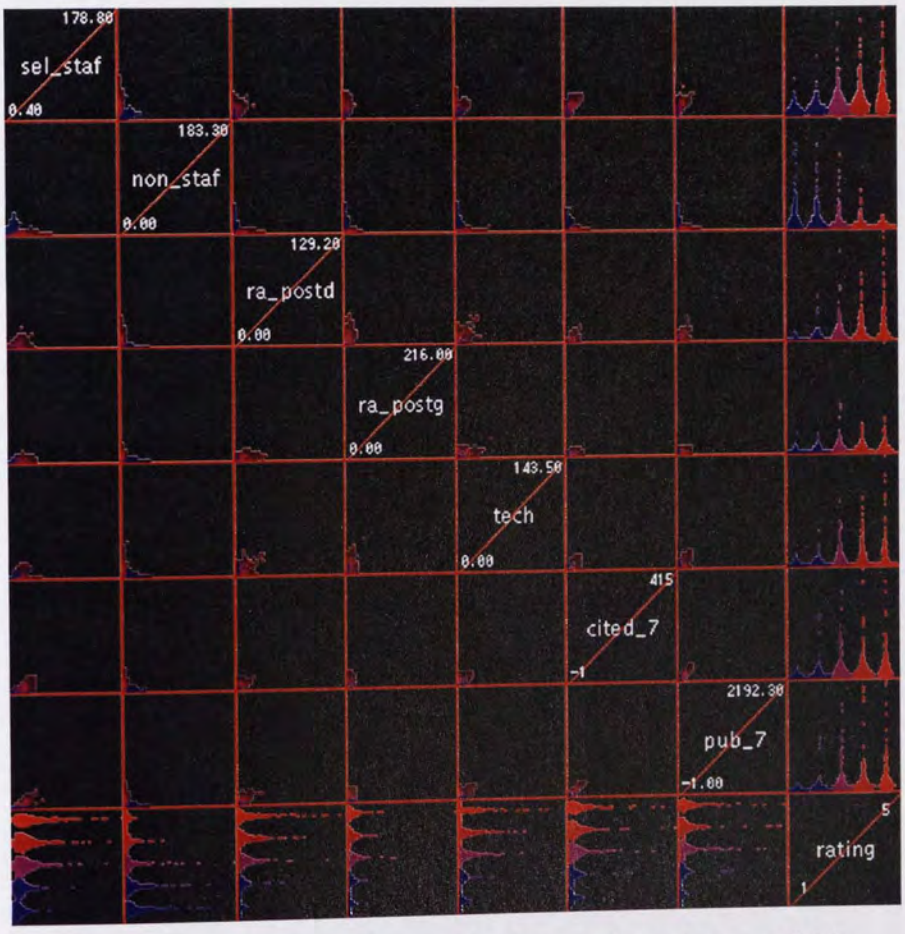


Plate 4.10 – 8 fields of the RAE database with `rating` overlaid

Plate 4.10 shows eight fields of the RAE database with the `rating` field overlaid using the standard blue-red colour scale. Two problems are apparent. Firstly, due to the high density areas near the origin of most plots, it is very hard to see the details in the low-intensity areas. Secondly, this colouring shows the low ratings in blue, the middle ratings in purple and the high ratings in red, which, though allowing the user to gain

a rough picture of where the highly-rated departments are, does not make it easy to identify, say, those departments which rated a four. These two problems were solved by introducing an alternative colour scale for overlays.

The intensity of the new overlay was kept constant, to prevent 'dim' areas on density plots (at the expense of losing the density information, but this can easily be recovered by removing the overlay). The problem with the choice of colours is that the overlay field is no longer a binary response, as it is in the mail and finance databases: there are five discrete 'responses' which, though ordered, are distinct. In order to enable the five different ratings to be clearly seen as an overlay, the new colour scale represents the five ratings as red, orange, green, light blue and dark blue respectively. Intermediate colours also exist in the scale, to allow overlapping points of differing ratings to be identified.

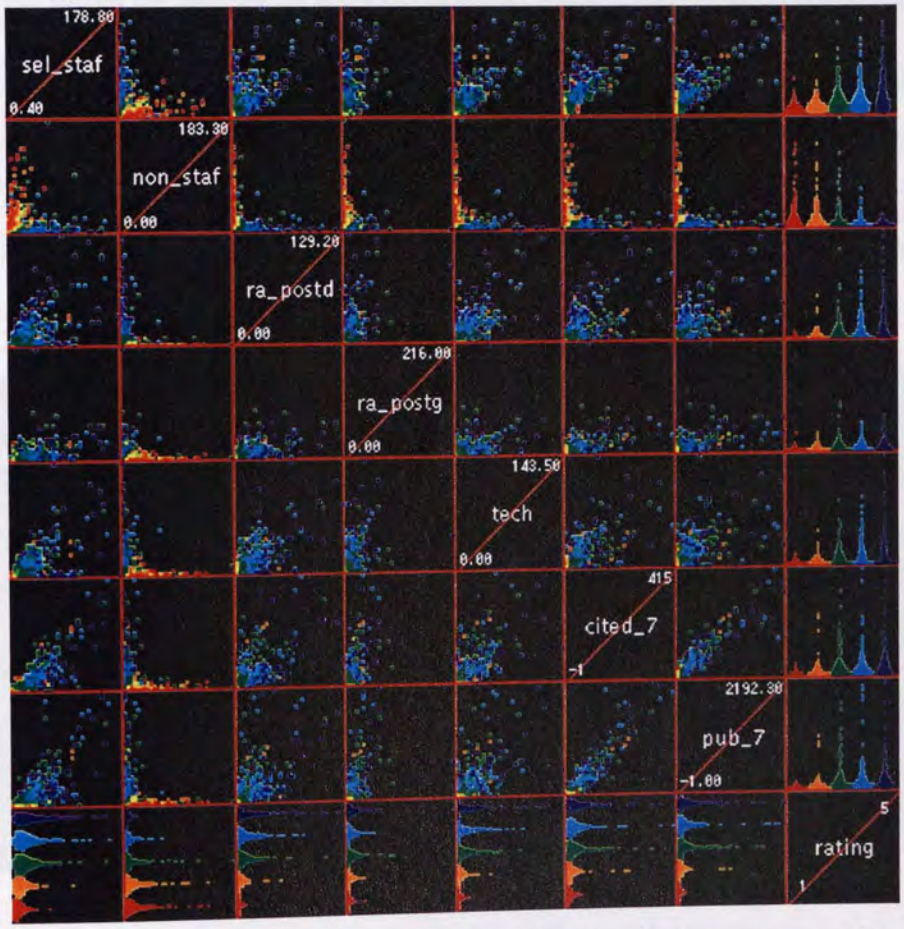


Plate 4.II – 8 fields of the RAE database with `rating` overlaid using new colour scale

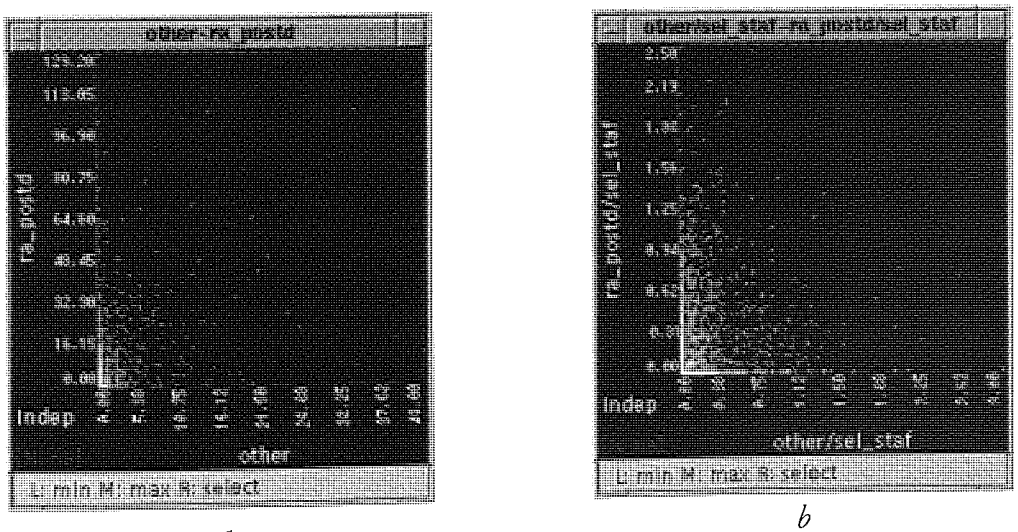
A button was added to top of the overview window to switch the alternate colour scale on and off. The result of using this new colour scale with the same overview as before is shown in plate 4.II. Clearly, both problems identified with the blue-red colouring have been solved, enabling exploration of the RAE database to proceed.

4.3.3.3 Field division

As noted in the previous chapter, many fields of the RAE database have a few very large values, which produce skewed plots. Also, figure 3.16*b* showed that the rating was very roughly proportional to the number of selected staff. Maybe if every field in the database could be normalised against `sel_staf`, some of the problems of skewed data would be alleviated.

A general-purpose 'divide' option was added to the axis menu (seen in figure 4.7 on page 101) which divides every field (except the field used as the divisor, categorical fields, and the final, response, field) in each record by the value of the specified field in that record, and creates a new overview containing these new fields together with the unchanged fields. The divided fields in the new overview are named accordingly, e.g. `tech/sel_staf`. Records where the divisor field is zero are omitted, and any unknown values (in the case of the RAE database, those set to -1) are untouched.

This is the first data processing operation to be introduced to the MADEN system. It modifies the data in an attempt to make it more informative. As an example of the use of the divide operation, the enlargement in figure 4.19*a* has the majority of its points near the origin, but following a division by `sel_staf`, the second enlargement (figure 4.19*b*) shows a reduction in the skewing.



a *b*

Figure 4.19 – Enlargements showing the effect of the divide operation

The divide operation is only of use in a few circumstances, but as part of the overall MADEN system it offers the user another tool with which to probe the data.

4.3.3.4 'Serial number' fields

Two of the fields in the RAE database are serial numbers, giving the number of the institution and the numeric 'unit of assessment'. Modifying these fields is pointless, so a method for preventing such division was needed.

Rather than make the program specific to this database, a general way to indicate that certain fields are serial numbers and are not to be altered by any processing was implemented by detecting the '#' character at the start of the field name. Thus the RAE database has two fields entitled `#inst` and `#uofa`.

The initial field choice routine was modified to ignore serial number fields, under the assumption that the user is unlikely to want to view such fields in an initial overview of a database.

This serial number mechanism proved to be of use when implementing more complex processing operations (as described in later chapters), and also for the next modification, highlighting.

4.3.3.5 Highlighting

When investigating the RAE database, it was often desired that the identity of particular data points could be determined – for example to discover which departments were exceptional. This was implemented using the right mouse button in enlargements. The density matrix is examined to identify whether there is a data point at, or in the near vicinity of, the click position. If there is, all the records which are projected to that data point in the density matrix are 'highlighted'. In order to enable the user to clearly see which element of the density plot will be highlighted, the cursor is changed to a crosshair when over an enlargement window.

Using the serial number information of the highlighted records, an external program is called to look up the name of the institution and unit of assessment and display these, along with the associated rating, in a separate window. In this way the user can observe an outlying data point, click on it, and (almost) immediately discover which department(s) it represents.

Highlighting is also shown graphically by drawing a small rectangle around each highlighted record on each overview density plot. For clarity, the rectangles are drawn in orange, unless there is an overlay, in which case white is used. The use of the overview highlighting enables the user to see whether an outlying point in one particular enlargement is an outlier in any of the other plots of the overview, for example.

For example, in the enlargement shown in plate 4.12, the green (i.e. rating 3) outlying point at the middle left is revealed to be Hospital Clinical at the University of Glasgow. Plate 4.13 demonstrates the overview highlighting resulting from clicking on this point, and shows that this department is fairly unusual in all the displayed fields.

Two more clicks disclose that the dark blue (i.e. rating 5) outlying point at the top of the plot is Education at the Institute of Education, and the light blue (i.e. rating 4) outlying point at the bottom right of the plot is Art & Design at the Royal College of Art.

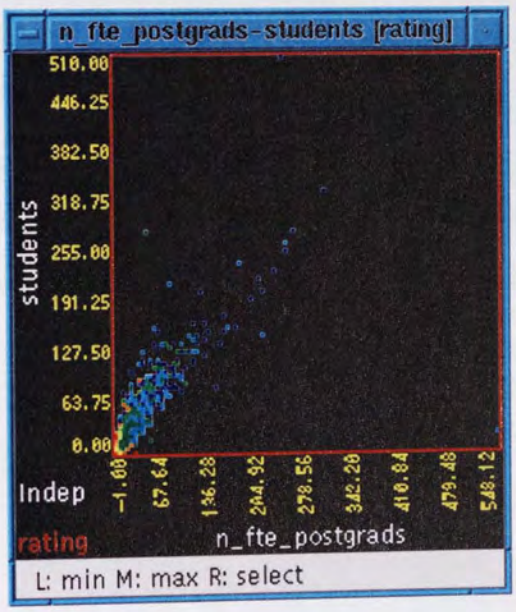


Plate 4.12 – Enlargement showing outliers which will be highlighted

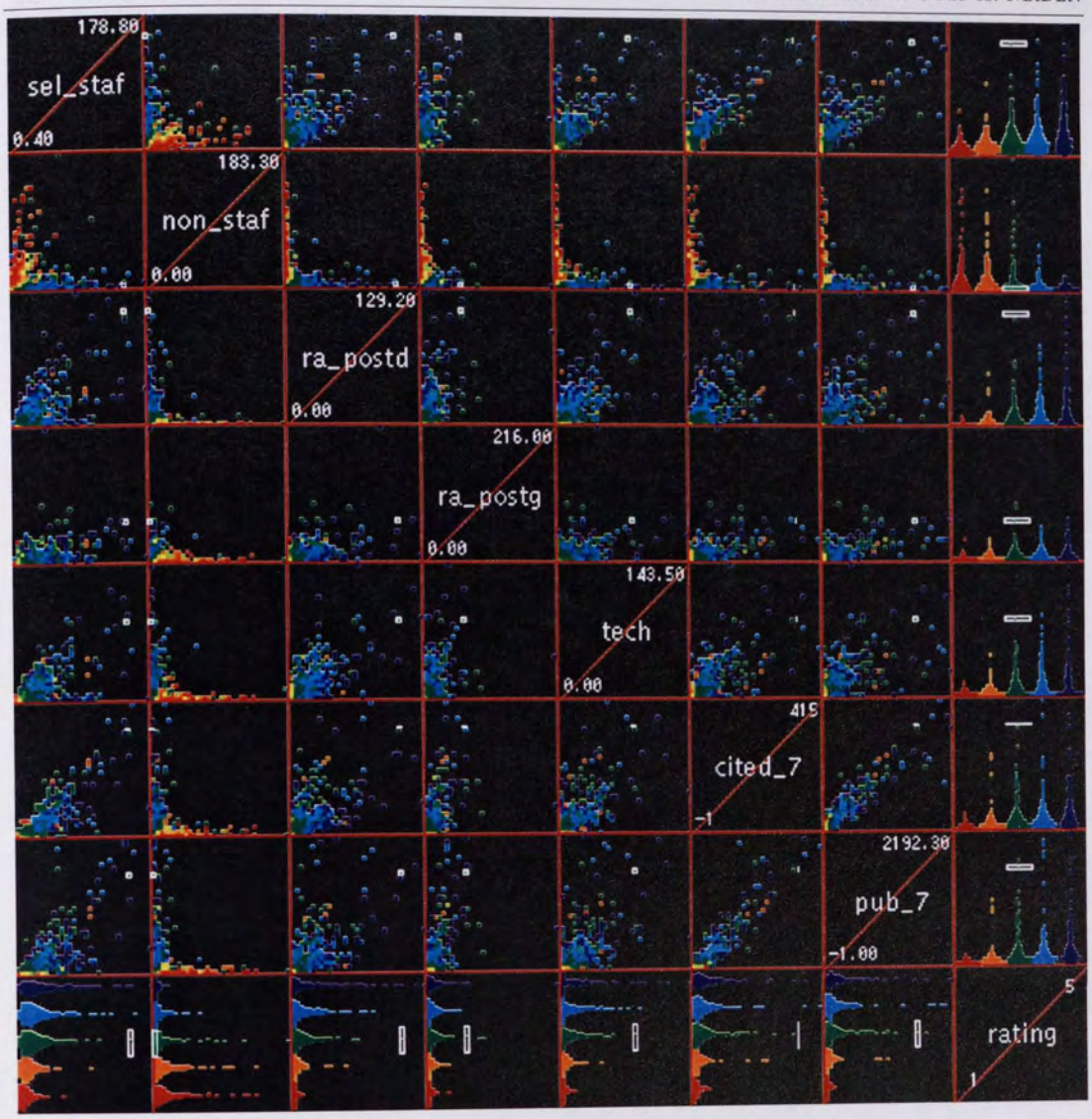


Plate 4.13 – Overview showing a highlighted data record

4.4 Conclusions

4.4.1 Concept

The MADEN system is a novel visualisation tool well suited to the investigation and exploration of tabular databases.

The use of Benediktine density plots gives the user a much clearer impression of the structure of the database under consideration than standard scatter plots, particularly with discrete fields, and the use of a coloured overlay adds a third dimension to each density plot

The overview window offers the possibility of viewing anything from two fields to the entire database in one window and visually identifying relationships between pairs of fields; the enlargement windows allow density plots to be expanded to examine 2-D relationships in detail.

n-dimensional selection using both the point-and-click and typed interfaces permits a hypercuboid of the database to be visually or numerically isolated, for example to extract an area of high response. Additionally, the selection can be used to control dependent enlargements, which further increase the power of the enlargement window.

The alternative colour scheme and highlighting mechanism provide specialised tools for investigating the RAE database, the latter being of general use for any databases with serial number information, such as customer ID numbers.

4.4.2 Implementation

MADEN overcomes many of the practical limitations of the Benediktine cell implementation described in chapter 3. In particular, the increased resolution of density plots, the axis labels, the improved response time and the more accessible user interface all contribute to make a much more usable and useful system.

Response time is generally adequate, in that selections can be modified and dependent enlargements updated fast enough to prevent users feeling they are being 'held back' – though not quite fast enough to allow continuous tracking of a changing selection (i.e. holding down the mouse button and dragging to watch the effect in real-time).

Changing the displayed fields in an overview can sometimes be rather slow, but not exceptionally so.

The ability to select only one category of a categorical field for use as an overlay expands the power of the overlay mechanism beyond that of the Benediktine cell implementation. As has been seen, the user is now able to identify precisely where customers who rent their homes are located, for example.

Selections, as in the cell, are constrained to be hypercuboidal, i.e. a simple range selection on every axis. However, extending the selections beyond this limit would be difficult and probably clumsy in use. A 'brushing' technique [Becker & Cleveland, 1987] as used in programs such as *xgobi* might seem feasible, but generally brushing is more suited to small databases where each record is visible and can be highlighted in some fashion. With a large database in MADEN, there is no obvious way in which large numbers of records could be highlighted successfully.

The user interface is more 'polished' than in the cell implementation. The screen is uncluttered by popup windows, although the displayed axes window is maybe not an ideal method of choosing axes. The field control windows offer a comprehensive set of tools for controlling selections, and the use of text in window footers to show the current effect of the mouse buttons provides the user with a helpful reference.

4.4.3 Evaluation

Figure 4.20 below repeats the evaluation chart given at the end of chapter two. It can be seen that MADEN fulfils all the criteria which none of the other techniques could match. In particular, it handles discrete data better than a standard scatterplot, it can display high-dimensional, large-sized databases with relative ease, it allows quantitative measurements from its displays, and it shows where records are coincident.

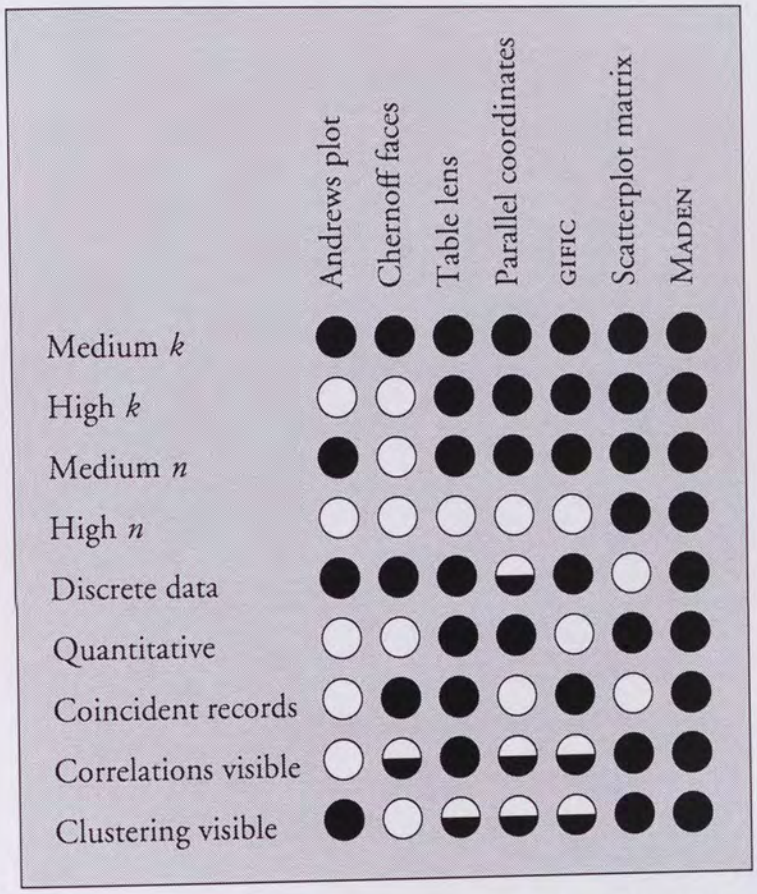
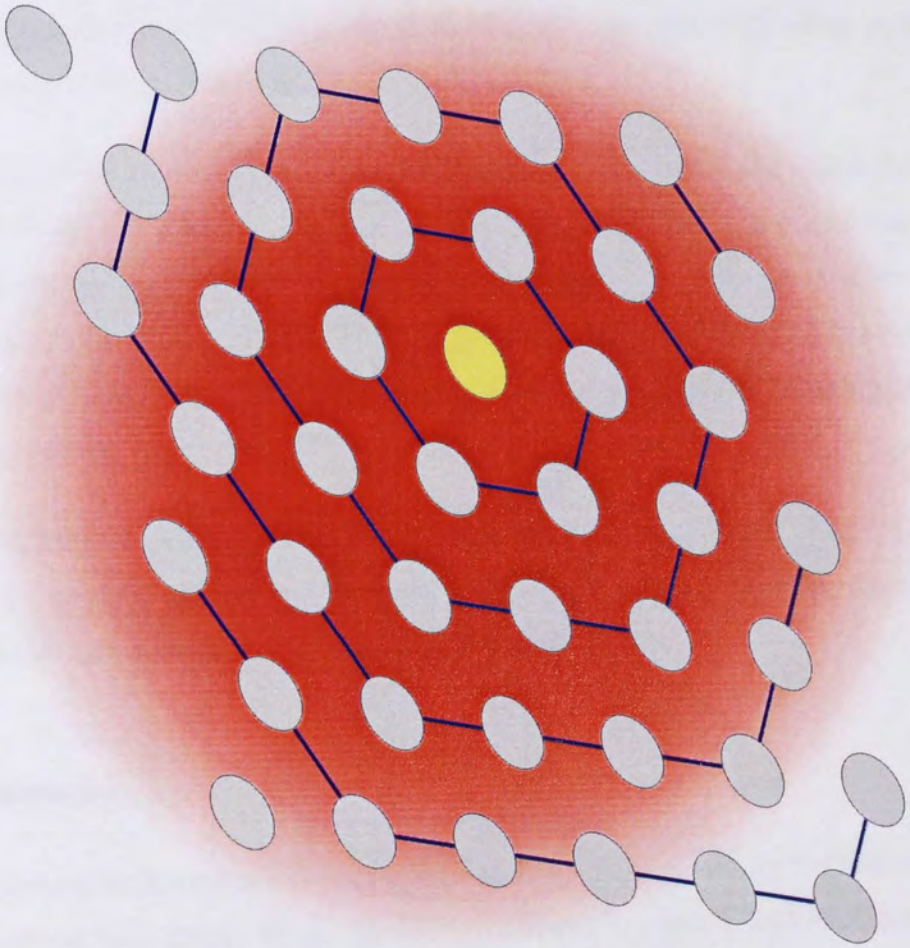


Figure 4.20 – Visualisation evaluation chart, including MADEN

It should be remembered that the evaluation below is based purely on the static display of MADEN (such as in figure 4.1 on page 95) – when the interactive features are taken into consideration, it is considerably more powerful, and even more so when used in conjunction with the many data processing operations which will be discussed in the remaining chapters of this thesis.

Chapter 5



Data Reduction: Clustering

J. J.
M. M.
W. G. Du P.
Took great
*C/o his M******
Though he was only 3.

[Milne, 1924]

5.1 Introduction

As has been seen, the sort of databases which the MADEN system is designed to visualise have many records. This leads to delays in generating and updating displays.

One way to improve the speed of the system would be to create a model of the database with less records than the database itself. This would reduce the number of records which had to be displayed, thereby reducing the time required to project and display a new view of the data.

There is, of course, a trade-off between the amount of information which the model retains about the database and the size of the model. What is required is a way of analysing the large number of records in the database in order to identify a small number of clusters into which the data is grouped. For example, a database consisting of two well-separated, self-contained groups of customers might be easily modelled by two clusters.

In addition to increasing the speed of the system (once the clusters have been found), a good clustering process will have the effect of 'smoothing' the data, removing the fine detail and allowing the overall shape therein to be seen.

5.1.1 Segmentation

The process of clustering identifies groups of data records which are located in similar areas in the data space. When applied to customer databases, this should naturally result in 'segmentation' of the customers:

> *For most marketing people segmentation is a commonly used and well understood concept. It can be defined broadly as the division of individuals by a characteristic such as their age or gender. The rationale behind segmentation is that people with one or more similar characteristics will share similar needs which can be satisfied by the same products or positioning of a product.* [Ballé & Jones, 1993]

Thus with a successful clustering algorithm, a marketing manager using MADEN should be able to find a set of natural segments in the particular database under scrutiny, rather than having to rely on predetermined segments which might not be applicable.

5.1.2 Requirements

Most of the clustering techniques which will be discussed in this chapter require three quite separate things: a distance measure, a clustering algorithm, and a display method.

5.1.2.1 Distance measure

In order to perform a clustering operation, a method for measuring a 'distance' between any two points in the data space is needed.

To be of practical use, this measure:

- should be unaffected by the scale of any field (i.e. in the mail database, an age difference of 80 should result in a much larger distance measure than a maximum balance difference of 80)
- should not impose an order upon unordered categories (i.e. in the mail database, the distance from a customer who owns his home to one who rents should be the same, all other fields being equal, as to one who lives with his parents)

5.1.2.2 Clustering algorithm

The clustering algorithm uses the distance measure to generate a set of clusters from the database. A suitable clustering algorithm must:

- generate a 'good' model of the data (i.e. one which models the features of the database as accurately as possible, given the operating parameters)
- produce a result in an acceptable time (there is no point generating clusters in an attempt to increase the speed of the system if the clustering process takes as long as the entire process would without clustering)
- use as few clusters as necessary (if surplus clusters are generated, they will merely slow the display down again – if it is possible to model the data using fewer clusters, the algorithm should do this)

Many common clustering algorithms (single linkage, complete linkage, group average, etc [Everitt, 1974]) require a *similarity matrix* to be constructed, which gives a measure of how similar each data point is to each other data point. This is an $O(n^2)$ process, and with the databases considered here, such a matrix would require vast amounts of memory and processing time to construct and search – e.g. a mere 10,000 records would require at least 49,995,000 floating-point matrix elements to be calculated and stored, before any cluster-finding process could even begin.

Techniques which are sub-$O(n^2)$ and consider records sequentially are thus required.

5.1.2.3 Display method

A set of clusters, once generated, must be presented to the user in some way. The display method should:

- show as much information as possible about the clusters
- while being displayed faster than the original data

5.2 Distance Measures

5.2.1 Notation

This chapter introduces mathematical notation for working with the databases. Some of the more common expressions are described below:

n	the number of records in the database
p	the number of fields in the database
$\mathbf{x}_i$	the i th record of the database (column vector)
$\mathbf{x}, \mathbf{y}, \mathbf{z}$	general data vectors
x^k	the (scalar) value of the kth field of data record $\mathbf{x}$
$width^k$	the range of the kth field across the entire database

5.2.2 Distance metrics

Many clustering algorithms require a distance measure which qualifies as a *distance metric.*

A metric $d(\mathbf{x}, \mathbf{y})$ which measures the distance between two data records $\mathbf{x}$ and $\mathbf{y}$ must satisfy the conditions shown below.

$$d(\mathbf{x}, \mathbf{y}) \geq 0$$
$$d(\mathbf{x}, \mathbf{y}) = 0 \Leftrightarrow \mathbf{x} = \mathbf{y}$$
$$d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$$
$$d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z}) \geq d(\mathbf{x}, \mathbf{z})$$

The first three of these conditions are generally met by any distance measure. The fourth, known as the 'triangle inequality', distinguishes distance metrics. As its name suggests, it constrains the distance between three points to behave in the same way as distances on a plane: the sum of two edges of a triangle always equals or exceeds the length of the third edge.

5.2.2.1 Euclidean metric

Probably the most common metric is the scaled Euclidean metric of equation 5.1.

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{k=1}^{p} \left(\frac{x^k - y^k}{width^k} \right)^2} \qquad 5.1$$

The difference between the coordinates on each axis is divided by the spread of values on that axis in order to make the metric independent of scale, and the standard Euclidean distance metric is then applied to these scaled differences.

5.2.2.2 City block metric

Equation 5.2 defines the scaled city block metric, which is simply the sum of the absolute scaled differences on each axis.

$$d(\mathbf{x}, \mathbf{y}) = \sum_{k=1}^{p} \left| \frac{x^k - y^k}{width^k} \right| \qquad 5.2$$

5.2.2.3 Minkowski metrics

A more general metric is the scaled Minkowski metric, shown in equation 5.3.

$$d(\mathbf{x}, \mathbf{y}) = \left\{ \sum_{k=1}^{p} \left| \frac{x^k - y^k}{width^k} \right|^r \right\}^{1/r} \qquad 5.3$$

With $r = 1$ this is the city block metric; $r = 2$ gives the Euclidean metric, which incidentally is the only rotationally-invariant case; as r increases further, d approaches the 'dominance metric' which is the largest single (scaled) difference on any axis.

The distance returned by the Minkowski metric ranges from zero to $p^{1/r}$. In order that the distance is unaffected by p (the number of fields), the distance equation is scaled by this factor, as shown in equation 5.4.

$$d(\mathbf{x}, \mathbf{y}) = \left\{ \frac{1}{p} \sum_{k=1}^{p} \left| \frac{x^k - y^k}{width^k} \right|^r \right\}^{1/r} \qquad 5.4$$

5.2.3 Differences between categorical fields

Measuring differences on continuous and integer fields is straightforward. However, for categorical fields, no ordering should be implied, so if the kth field is a categorical field, equation 5.5 can be used to find the 'difference':

$$\left|p^k - q^k\right| = \begin{cases} 0 & p^k = q^k \\ d_c & p^k \neq q^k \end{cases} \qquad 0 < d_c \leq 1 \qquad 5.5$$

d_c is the constant distance between non-identical categorical values – if it is unity, two non-identical categories are as far apart as possible, if it is zero, there is no difference between any categories. d_c may also be divided by the number of categories ($width^k$), so that the difference between binary categories is more than that between one-of-twenty.

5.2.4 Choice of measure

For maximum flexibility, the measure used in the clustering algorithms was the Minkowski metric, scaled as shown in equation 5.4, with the categorical difference modifications from equation 5.5. Suitable parameter values were found to be $r = 2$, $d_c = 0.5$, giving a Euclidean metric with a moderate difference between non-identical categories.

5.3 Sequential Leader Algorithm

A relatively simple sequential clustering algorithm is the sequential leader [Hartigan, 1975; Aldenderfer & Blashfield, 1984]. It is a very fast method, requiring only one pass through the database, and so it is roughly $O(n)$.

5.3.1 Algorithm

The clustering is performed by maintaining a list of cluster 'seeds' around which clusters form, and assigning each data point to one cluster. Specifically:

- Assign the first data point to be the first seed.
- For each remaining data point:
  - Find the first seed which is closer than a given threshold distance d_T from the point (d_T specifies the maximum size of each cluster).
  - If a seed is found, assign the point to its cluster and update the seed's position to be the mean of all points in its cluster.
  - If there's no seed within the threshold distance, create a new seed at the same location as the point, and assign the point to its cluster.

Two alterations were made to the algorithm: selecting the data points in a random order, in an attempt to reduce the effects of any ordering in the database, and choosing the closest (rather than the first) seed which is within the threshold distance.

5.3.2 Performance

Figure 5.1 shows the results of applying the sequential leader algorithm to the three databases. Seven different values of d_T were used, and the number of clusters generated was recorded. Also, the number of calls to the distance measuring function was noted, to give an indication of the processing time required.

The table shows that the number of clusters found is very sensitive to changes in d_T. A change of 0.025 in d_T roughly doubles the number of clusters, and also doubles the processing time required.

d_T	mail database		finance database		RAE database	
	clusters	distances	clusters	distances	clusters	distances
0.400	5	46,412	5	46,604	6	10,025
0.375	8	78,120	7	67,852	7	12,307
0.350	16	144,466	12	108,601	10	18,390
0.325	30	251,909	28	220,453	12	22,900
0.300	66	531,278	55	433,811	17	29,048
0.275	135	1,081,684	129	937,312	26	39,615
0.250	290	2,224,745	306	2,194,157	38	59,933

Figure 5.1 – Performance of the sequential leader algorithm on real data

5.3.3 Assessment

5.3.3.1 Model accuracy

Since there is no upper limit to the number of clusters, a cluster should exist within d_T of every data point. This results in a model at least as accurate as a set of hyperspheres of radius d_T. However, as we have seen, the number of clusters, and hence the quality of the model, is highly dependent on the value of d_T.

5.3.3.2 Speed

The process is fast, as it only makes one pass through the data, and one pass through the seeds for each data point.

5.3.3.3 Number of clusters

The number of clusters generated is highly dependent on d_T. Too large and only a few clusters are found; too small and too many clusters result. It appears that a value of around 0.325 gives acceptable results with the three example databases, though there is no guarantee that this would hold for other databases.

5.4 FASTCLUS

The FASTCLUS algorithm [SAS Institute Inc, 1988] is a hybrid of several clustering methods. It is used in the SAS/STAT statistics package.

5.4.1 Algorithm

The procedure is a development of Hartigan's, and is very similar to the standard 'K-means' algorithm [MacQueen, 1967], differing only in the initial step:

1 Select a set of data points to use as initial seeds. The algorithm for selecting seeds is complex and will not be detailed here; the seeds are carefully chosen to give an even spread through the database, no seed is closer than a given distance $d_{\min}$ from any other, and the number of seeds will not exceed a specified maximum $q_{\max}$.

2 Assign each data point to the nearest seed, forming temporary clusters.

3 Replace each seed by the mean of its temporary cluster.

4 Repeat steps 2 and 3 until there is no further change in seed positions.

As given above, the process takes a long time to converge, so the termination criterion (step 4) was modified to repeat until the average change in seed position is less than a given value δ_T. After convergence, any seeds to which no data points have been assigned are deleted.

There are three parameters: the minimum distance between seeds ($d_{\min}$), the maximum number of seeds ($q_{\max}$), and the threshold for the average change in seed position beyond which the process should stop (δ_T).

5.4.2 Results

Following some initial experimentation, the parameters $q_{\max}$ and δ_T were set to $10n^{1/5}$ and 0.05 respectively. This gave a manageable maximum number of clusters and a reasonably short processing time to optimise their positions. Figure 5.2 shows the number of clusters generated for various values of $d_{\min}$. It should be noted that $q_{\max}$ for both the mail and finance databases is 63. The number of calls to the

distance function were not recorded, as the number of iterations of the FASTCLUS procedure varied quite widely.

$d_{\min}$	mail database	finance database	RAE database
0.35	15	7	3
0.30	54	38	5
0.25	63	63	11
0.20	63	63	25

Figure 5.2 – Performance of the FASTCLUS algorithm on real data

The table shows that reducing $d_{\min}$ increases the number of clusters to a similar extent that reducing d_T does for the sequential leader algorithm. However, in this case the number of clusters is not allowed to exceed $q_{\max}$.

5.4.3 Assessment

5.4.3.1 Model accuracy

The seed selection procedure ensures that the available clusters are well-distributed throughout the database. Outlying points may well be contained in a cluster of their own, as they always are with the sequential leader algorithm, but may alternatively be included in a more distant cluster.

5.4.3.2 Speed

FASTCLUS is considerably slower than the sequential leader algorithm, due to the iteration involved. However, results are obtained in a couple of minutes on a loaded workstation, which is bearable.

5.4.3.3 Number of clusters

The major advantage that FASTCLUS has over the sequential leader algorithm is that the number of clusters is not allowed to grow in an uncontrolled fashion. The upper limit of $q_{\max}$ prevents unmanageable quantities of clusters, though with a suitable value of $d_{\min}$, this limit is not reached.

5.5 Mixture Models

A *mixture model* [Silverman, 1986; Everitt & Dunn, 1991; Scott, 1992] attempts to construct a model of the *distribution* from which the data might have been generated, by using a mixture of a small number (q) of simple distributions. The model investigated here uses a mixture of gaussian (normal) distributions.

Each gaussian $j = 1 \ldots q$ is defined by its centre $\boldsymbol{\mu}_j$ and a set of variances. There were three options for the variances:

- One scalar variance, applied to all axes of the data space, resulting in spherical gaussians. This was not considered powerful enough to model complex data.
- A different scalar variance for each axis, resulting in easily-visualisable axis-aligned elliptical gaussians.
- A full covariance matrix, resulting in completely general elliptical gaussians, which would be somewhat harder to display.

The second option was chosen, as it offered moderate modelling power together with the potential for straightforward visualisation of the resulting model. The standard deviation of the jth gaussian in the direction of the kth axis is then σ_j^k.

To complete the definition of the distribution, a set of 'mixing proportions' π_j are defined. π_j is the probability that a data point generated by the distribution will come from the jth gaussian.

5.5.1 Algorithm

The expectation maximisation (EM) algorithm was used to optimise the mixture model, as described in numerous textbooks [Titterington *et al*, 1985; McLachlan & Basford, 1988; Everitt & Dunn, 1991], adapted for axis-aligned elliptical gaussians with assistance from Chris Williams.

5.5.1.1 Initialisation

The initial centres are located on random non-coincident data points, with the initial standard deviations of each gaussian set to the distance to the nearest other centre.

The initial mixing proportions are made equal, setting them to the reciprocal of the number of gaussians in the mixture.

The input data is standardised to zero mean, unit variance, to simplify calculations and prevent large-ranged axes having more influence than small-ranged ones.

5.5.1.2 Expectation step

The 'responsibilities' are calculated as shown in equation 5.6. The responsibility r_{ij} is the probability that the ith data point was generated by the jth gaussian. The denominator of the equation ensures that the probabilities for each data point sum to unity.

$$r_{ij} = \frac{\dfrac{\pi_j}{\prod_{k=1}^{p}\sigma_j^k}\exp\left[-\sum_{k=1}^{p}\dfrac{(x_i^k-\mu_j^k)^2}{2(\sigma_j^k)^2}\right]}{\sum_{l=1}^{q}\left(\dfrac{\pi_l}{\prod_{k=1}^{p}\sigma_l^k}\exp\left[-\sum_{k=1}^{p}\dfrac{(x_i^k-\mu_l^k)^2}{2(\sigma_l^k)^2}\right]\right)} \qquad 5.6$$

5.5.1.3 Maximisation step

Having calculated the responsibilities, the centres, standard deviations and mixing proportions are updated using the formulæ shown in equations 5.7, 5.8 and 5.9 below. Note that the μ_j^k in equation 5.8 is the newly-updated μ_j^k from equation 5.7.

$$\boldsymbol{\mu}_j = \frac{\sum_{i=1}^{n} r_{ij}\mathbf{x}_i}{\sum_{i=1}^{n} r_{ij}} \qquad 5.7$$

$$(\sigma_j^k)^2 = \frac{\sum_{i=1}^{n} r_{ij}(x_i^k-\mu_j^k)^2}{\sum_{i=1}^{n} r_{ij}} \qquad 5.8$$

$$\pi_j = \frac{1}{n}\sum_{i=1}^{n} r_{ij} \qquad 5.9$$

The process then repeats – by recalculating the responsibilities using the new centres, standard deviations and mixing proportions in equation 5.6 – until the algorithm has converged to a stable model.

5.5.1.4 Likelihood

In order to measure the convergence of the algorithm, the 'likelihood' is calculated. This is a measure of the likelihood that the model could have been used to generate the data, and should always increase under the EM algorithm. Equation 5.10 shows the measure used.

$$L = \sum_{i=1}^{n} \log \sum_{j=1}^{q} \pi_j r_{ij} \qquad 5.10$$

5.5.2 Results

In use, several problems were found:

- Gaussians often 'collapsed' along one or more axes, leading to errors in equation 5.6. This was due to the discrete nature of many of the fields in the databases. To overcome this, a minimum standard deviation was defined (typically 0.005) and any σ_j^k which fell below this threshold was reset to this value.

- Mixing proportions also often became infinitesimal. In this case, it was assumed that the mixture did not require the gaussian in question, and it was removed from the mixture.

- The initial, random, choice of centres was frequently poor. To improve the selection of initial centres, the first part of the FASTCLUS algorithm was used. This resulted in slightly longer preparation time, but a much improved optimisation time, with fewer iterations being required before convergence.

- Evaluation and convergence, even after the above modification, were both very slow – optimising 50 clusters in the mail database took an entire morning.

5.5.3 Assessment

5.5.3.1 Model accuracy

The mixture model, in attempting to generate a model of the distribution of the data, should be expected to give a better model than simple clustering methods which merely move centres around and assign points to them.

Theoretically, the algorithm converges on the best possible model using a fixed number of axis-aligned elliptical gaussians.

In tests, with a low-dimensional database, the mixture model did indeed generate very good sets of clusters which accurately represented the shape of the data – as demonstrated by the illustrations in the next section.

5.5.3.2 Speed

Unfortunately, the algorithm performs extremely slowly when optimising cluster positions in a large, high dimensional database.

5.5.3.3 Number of clusters

Like FASTCLUS, the mixture model as implemented allows a maximum number of clusters to be specified. This maximum will not in fact be used if a simpler model is found to be possible.

5.5.4 Alternative approaches

A technique to aid the application of mixture models to the clustering of mixed-mode data (i.e. data which contains ordinal and/or binary fields) has been discussed [Everitt, 1988; Everitt & Merette, 1990]. The approach is to assume that the non-continuous variables are the result of thresholding a set of unobserved continuous variables.

However, this method results in multidimensional integrals which make optimisation 'computationally no easy task' [Everitt, 1988], particularly when the data has a large number of non-continuous variables – as is the case with the databases considered here. For this reason, the mixed-mode approach was not implemented.

5.6 Choice of Algorithm

The sequential leader algorithm, though fast, requires careful setting of its threshold parameter to prevent the generation huge numbers of clusters. Also, it can be considered to be a special case of the FASTCLUS algorithm. It was therefore rejected.

The choice was then between FASTCLUS and the mixture model. The mixture model has the clear advantage that its clusters are of definite shape – indeed, an easily-visualised elliptical shape.

The mixture model clustering algorithm was thus chosen for use in MADEN.

5.6.1 Implementation

A 'Cluster' button was added to the top of the overview window. When pressed, the clustering routine is called, and a new overview is created to display a reduced database, in the form of the resultant clusters. How the clusters are shown in this overview is the subject of the next section.

5.7 Display Techniques

In order to illustrate the development of the display methods in this section, enlarged projections of the same set of clusters will be used throughout. The plot is of `Age` against `Ac_Turn` from the mail database. The mixture model algorithm was used to generate twenty clusters from a 3-D space containing `Age`, `Ac_Turn` and `Maxbal`.

For reference, the projection before clustering is shown in figure 5.3:

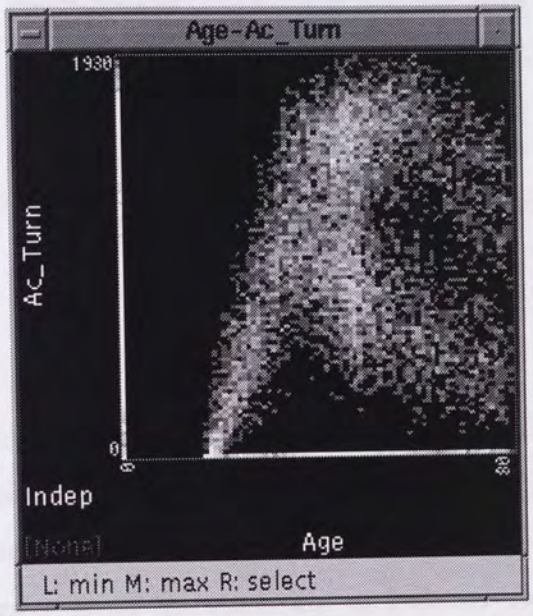


Figure 5.3 – Enlarged view of the projection before clustering

5.7.1 Centres

The simplest and fastest display method is to merely plot a point (actually two pixels square, to increase visibility) at the centre of each cluster. Its colour can be varied, in order to indicate the number of data points contained within the cluster – i.e. the density of data in the cluster. The chosen method of showing the density is to shade the plotted point with a greyscale, so that sparsely populated clusters appear dimmer than dense ones. Figure 5.4 overleaf shows the centres of the demonstration clusters.

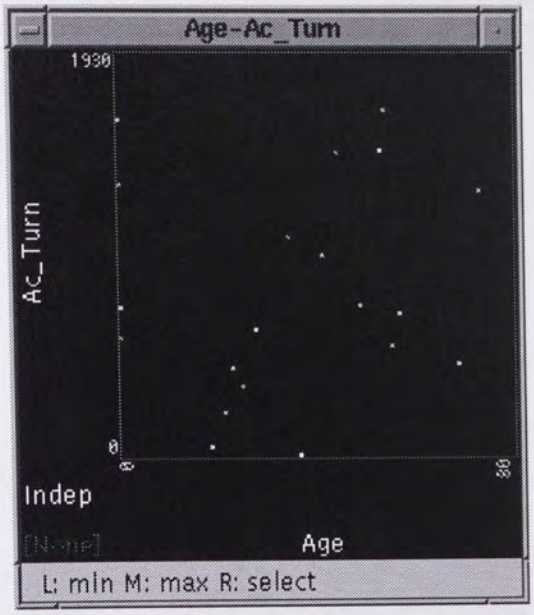


Figure 5.4 – Cluster display showing centres

5.7.2 Ellipses

The gaussian mixture model was chosen in order to generate axis-aligned ellipses. These ellipses can easily be drawn to give an indication of the size of each cluster. An ellipse is shown at the standard deviations of the projection of the cluster on to the appropriate axes, again using the greyscale to indicate the number of records encompassed by the cluster, as shown in figure 5.5.

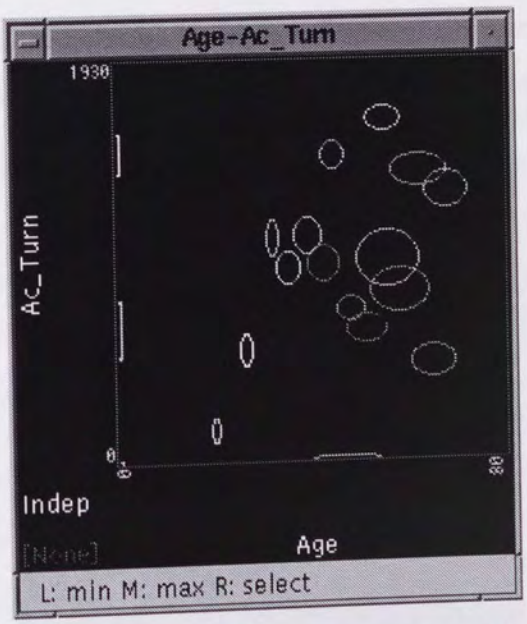


Figure 5.5 – Cluster display showing ellipses at the standard deviation of the clusters

With little loss of speed, the ellipses may be filled with the greyscale colouring. However, clusters which are plotted later mask those underneath. To overcome this,

the clusters are ordered by density before plotting, so brighter shapes mask only dimmer ones beneath. Figure 5.6 shows the resulting display.

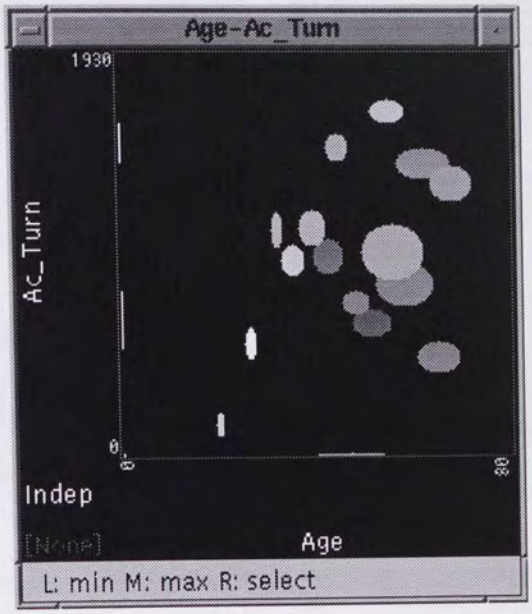
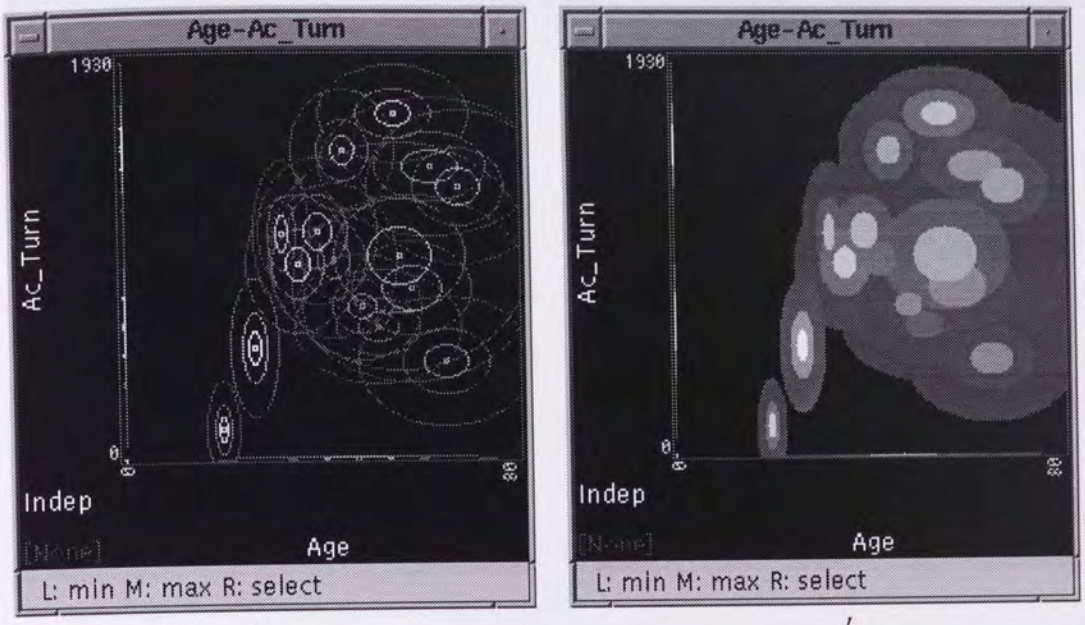


Figure 5.6 – Cluster display showing filled ellipses

5.7.3 Multiple ellipses

As figures 5.5 and 5.6 show, ellipses drawn at the standard deviation are not large enough to make the cluster display appear similar to the original density plot (figure 5.3). Experiments were carried out with ellipses of larger size, and as a result it was decided to allow the user to choose to display ellipses at the standard deviation and at two and four times this distance. Also, the display of centre points was made optional.

Figure 5.7 – Cluster display showing three ellipses per cluster



a *b*

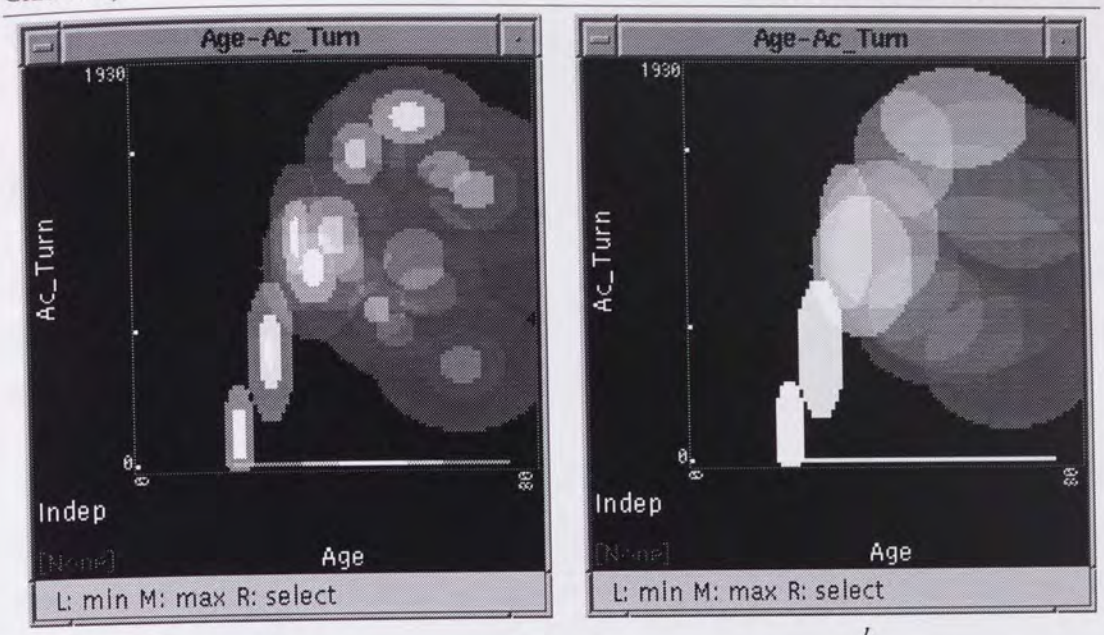
Figure 5.8 – Cluster displays showing multiple ellipses

Figures 5.7 and 5.8 show the results of using multiple ellipses. Figure 5.7 shows the cluster centres with outlined ellipses drawn at one and two standard deviations; figure 5.8*a* adds the outline at four standard deviations. Figure 5.8*b* uses filled ellipses at one, two and four standard deviations. Once again, the order of drawing the sixty ellipses (twenty clusters with three ellipses each) has to be carefully planned to minimise the amount of information which gets hidden.

5.7.4 Density plots

In an attempt to overcome the loss of information caused by drawing later, denser, clusters over previous ones, an additive plotting mechanism was developed. A density matrix is built up by calculating which elements are covered by each ellipse and incrementing them accordingly. It is the displayed using the existing greyscale density matrix plotting routines.

Figure 5.9 overleaf shows examples of clusters shown using density plots. Figure 5.9*a* has ellipses at one, two and four standard deviations; figure 5.9*b* shows only the ones at four standard deviations. The effect of building up the density can clearly be seen.



a *b*

Figure 5.9 – Cluster displays showing density plots

Though density plotting generates by far the best representation of the clusters, it is a very slow process, since each displayed point in each ellipse requires an increment in the density matrix. As can be seen in figure 5.9, the matrix is constructed at half the resolution of the display (compare with figure 5.8, which uses the maximum resolution), but even with this measure, the speed is a lot slower than displaying density plots of the original data.

5.7.5 Conclusions

As the figures in the section have demonstrated, the use of filled or outlined ellipses positioned at multiples of the standard deviations of the clusters gives a clear picture of the position of the clusters. The method is fast and versatile, and generally very suitable for cluster visualisation in MADEN.

The density plot technique, as has been noted, gives a more informative display, but is considerably slower than 'flat' plots. Indeed, it is slower than projecting and displaying the original data, which would suggest that it is of little practical use, except when visualising an exceptionally large database.

5.8 The Kohonen Self-Organising Map

5.8.1 Introduction

A different approach to clustering the data is to use a Kohonen self-organising map [Kohonen, 1990], which is based on ideas of how certain parts of the human brain learn to recognise patterns.

This method is similar to the previous ones in that it attempts to optimise the position of centres in the data space onto which all the points in the database are mapped, but it has an important difference which enables the map to model some of the topology of the data.

5.8.2 Structure

The Kohonen map is comprised of a regular lattice of 'nodes'. The lattice is usually rectangular or hexagonal, so that each non-edge node has four or six immediate neighbours respectively.

For MADEN, a hexagonal lattice was chosen, as it is more easily seen as a connected lattice than a square one [Carr, 1991]. The lattice structure is shown in figure 5.10, which also shows the rectangular coordinate system used to store the hexagonal lattice in a rectangular matrix. The lattice is made non-square in size to prevent the symmetry of a square layout which might prevent the map from reaching a good solution [Kohonen *et al*, 1995].

Each node i has a 'weight vector' $\mathbf{m}_i$ which is a location in the data space, corresponding to the 'centre' of its cluster.

5.8.3 Training algorithm

Training consists of repeatedly applying input vectors to the map. The distance between the input vector and each weight vector is calculated (using a suitable metric), and the node with the smallest distance is declared to be the 'winning' node. The weight vectors of the winning node and, crucially, other nodes in its 'neighbourhood' are then moved fractionally closer to the position of the input vector.

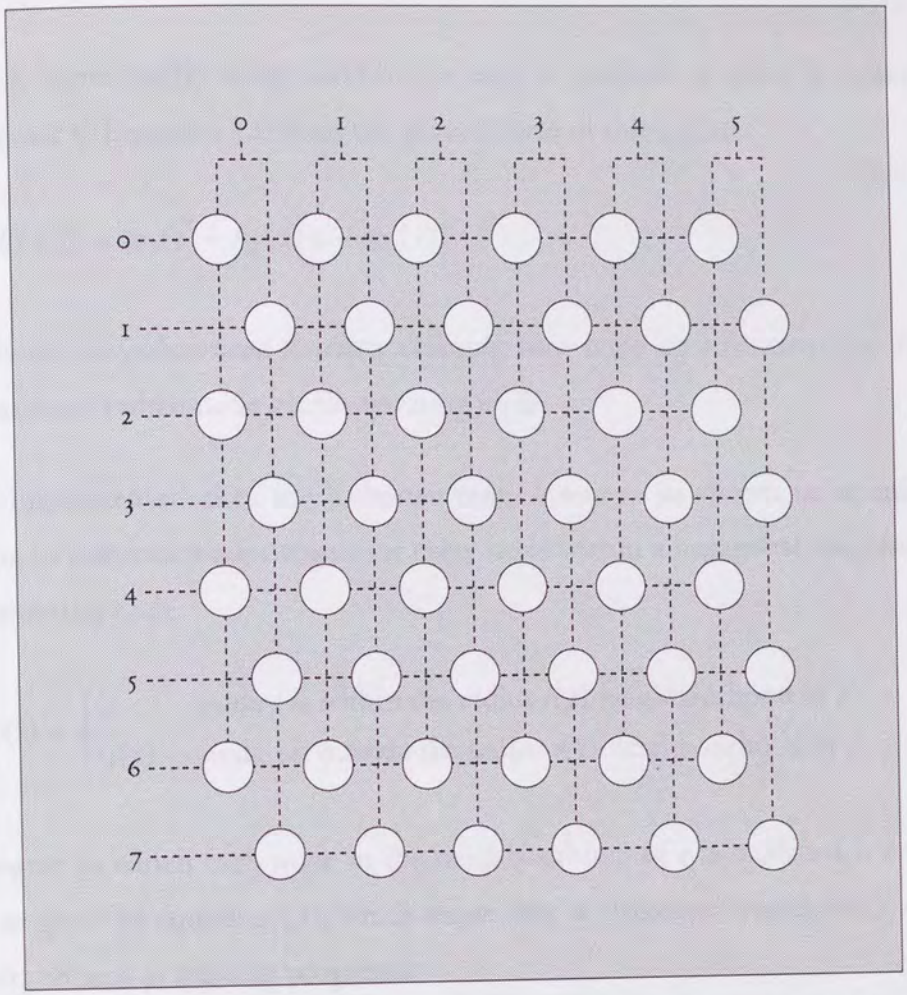


Figure 5.10 – Structure of the Kohonen map, showing the coordinate system

Mathematically, the training algorithm is as follows:

1 Initially, the data is standardised and each node's weight vector $\mathbf{m}_i$ is set to a random value.

2 A random data point $\mathbf{x}$ is chosen, and the winning node c whose weight vector $\mathbf{m}_c$ is closest to $\mathbf{x}$ is found.

3 Every weight vector $\mathbf{m}_i$ is then updated (as described below).

4 Steps 2 and 3 are repeated for a given number of cycles T. As a measure of how well the map fits the data, the average distance between $\mathbf{x}$ and $\mathbf{m}_c$ is monitored.

5.8.3.1 Update equations

In step 3, (potentially) every node in the map is updated to move it closer to the input vector **x**. Equation 5.11 gives the general form of this update.

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + h_{ci}(t)\left[\mathbf{x} - \mathbf{m}_i(t)\right] \qquad 5.11$$

$h_{ci}(t)$ is the *neighbourhood function*, defining how large an adjustment to make to node i at time t when node c is the winning node.

In this implementation of the Kohonen map, h was set as shown in equation 5.12, resulting in a constant adjustment for every node within a hexagonal neighbourhood of the winning node.

$$h_{ci}(t) = \begin{cases} 0 & \text{node } i \text{ is within the radius } r(t) \text{ neighbourhood of } c \\ \alpha(t) & \text{node } i \text{ is outside the radius } r(t) \text{ neighbourhood of } c \end{cases} \qquad 5.12$$

The degree to which each node in the neighbourhood of c is modified is controlled by α, as given by equation 5.13, which shows that α decreases linearly from an initial value α_0 to zero as training progresses.

$$\alpha(t) = \alpha_0\left(1 - \frac{t}{T}\right) \qquad 5.13$$

The neighbourhood is defined as the hexagon of radius r, as demonstrated by figure 5.11, which shows the map with node (3,2) active. The numbers on the nodes show their radius from the winning node c, and the black rings show the extent of each neighbourhood. The radius r is dependent on t, decreasing linearly from an initial value (r_0) to one as the training progresses (shown in equation 5.14).

$$r(t) = r_0 - (r_0 - 1)\frac{t}{T} \qquad 5.14$$

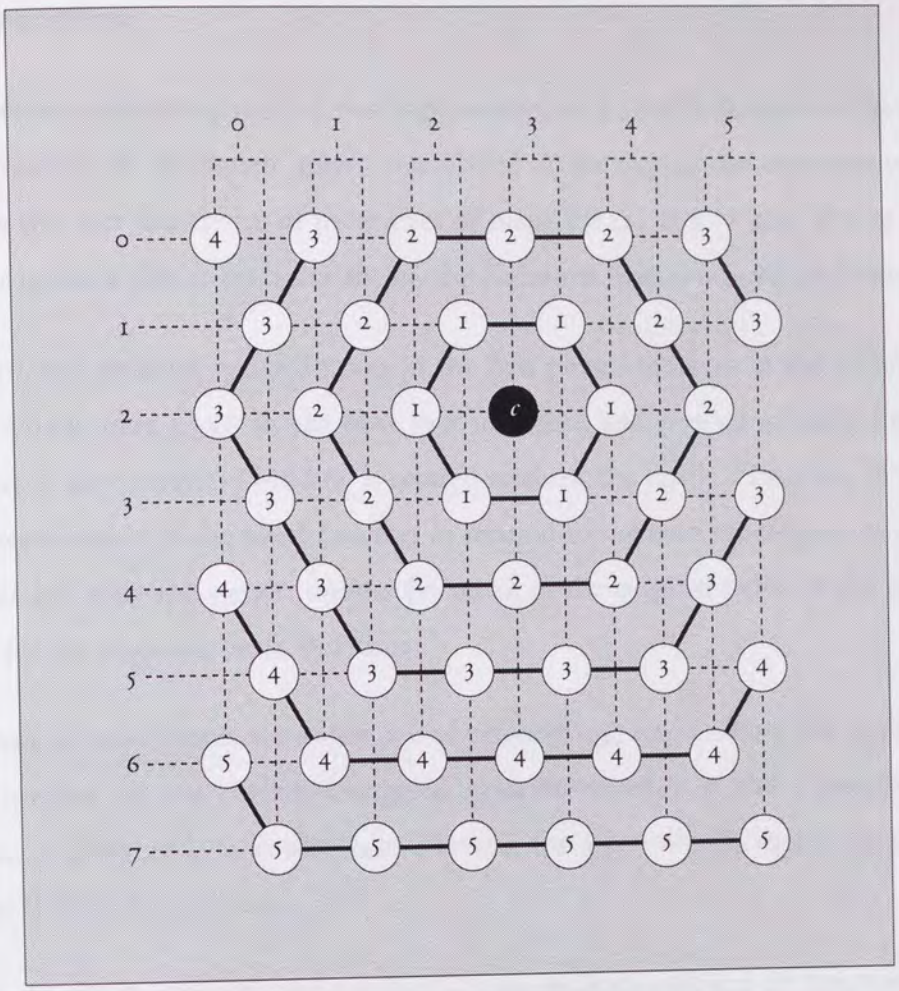


Figure 5.11 – The Kohonen map, showing the winning node c at (3,2) and neighbourhood radii on other nodes

5.8.4 Training process

Training consists of two phases. In the first phase, r_0 is set so large that the whole map is initially modified during each update, and α_0 is set to 0.05 to ensure that the distance by which each weight vector moves is considerable. This phase has the effect of moulding the map to the general shape of the data, stretching itself in the multidimensional space.

The second phase sets r_0 to a smaller value (3), to localise the effects of updates, and α_0 is reduced to 0.02 to reduce the impact of each update. This phase lasts ten times as long as the first phase, and has the effect of finely-tuning the weight vectors to model the shape of the data as closely as possible.

The figure for r_0 and α_0 are taken from the description of the 'official' Kohonen map code, SOM_PAK [Kohonen *et al*, 1995].

5.8.5 Implementation

The Kohonen clustering routine was implemented in a similar fashion to the mixture model routine. A 'Kohonen' menu was placed at the top of the overview window, offering the user the choice of three sizes of map: 8×12, 13×17 and 18×22. When the user makes a choice from this menu, the Kohonen map is created and trained.

After 110,000 training cycles (10,000 in the first phase, 100,000 in the second – the values which were given in the SOM_PAK literature and proved suitable for use in MADEN), a new overview window is created to show the result. However, in order to allow examination of the weight vectors in relation to the map topology, this overview contains not only the weight vectors (in terms of the original fields of the database, except for the response field), but also:

- a pair of coordinates, Kohonen_x and Kohonen_y, which allow the nodes to be placed on the correct hexagonal grid. Kohonen_y is the y coordinate shown in figure 5.10; Kohonen_x is double the x coordinate, incremented by one if Kohonen_y is odd.

- a field which contains the number of data records assigned to the node in question, named Kohonen_den as it conveys density information.

- a field called Separation which contains the distance to the furthest neighbouring node. This should allow clustering in the map to be visualised – where there is a jump from one cluster to another, the distance will be large.

The user is now able to open an enlargement window showing the two coordinate axes, and overlay any of the other fields on it. For example, overlaying a field of the weight vector demonstrates how its component changes across the map, as will be seen in plate 5.1 on page 161. With a little concentration, the hexagonal map structure, though not immediately clear, can be seen within the rectangular plot.

5.9 Use with Real Data

5.9.1 Mail database

5.9.1.1 Mixture model

The examples of mixture model clusters shown earlier in this chapter were generated from a three-field version of the mail database. Performing clustering on the entire mail database to generate fifty clusters was not as successful. As figure 5.12 shows, the clusters retain very little of the original structure of the database. Most clusters are centred near the middle of the fields, and are relatively wide – the ellipses in the figure are at one standard deviation.

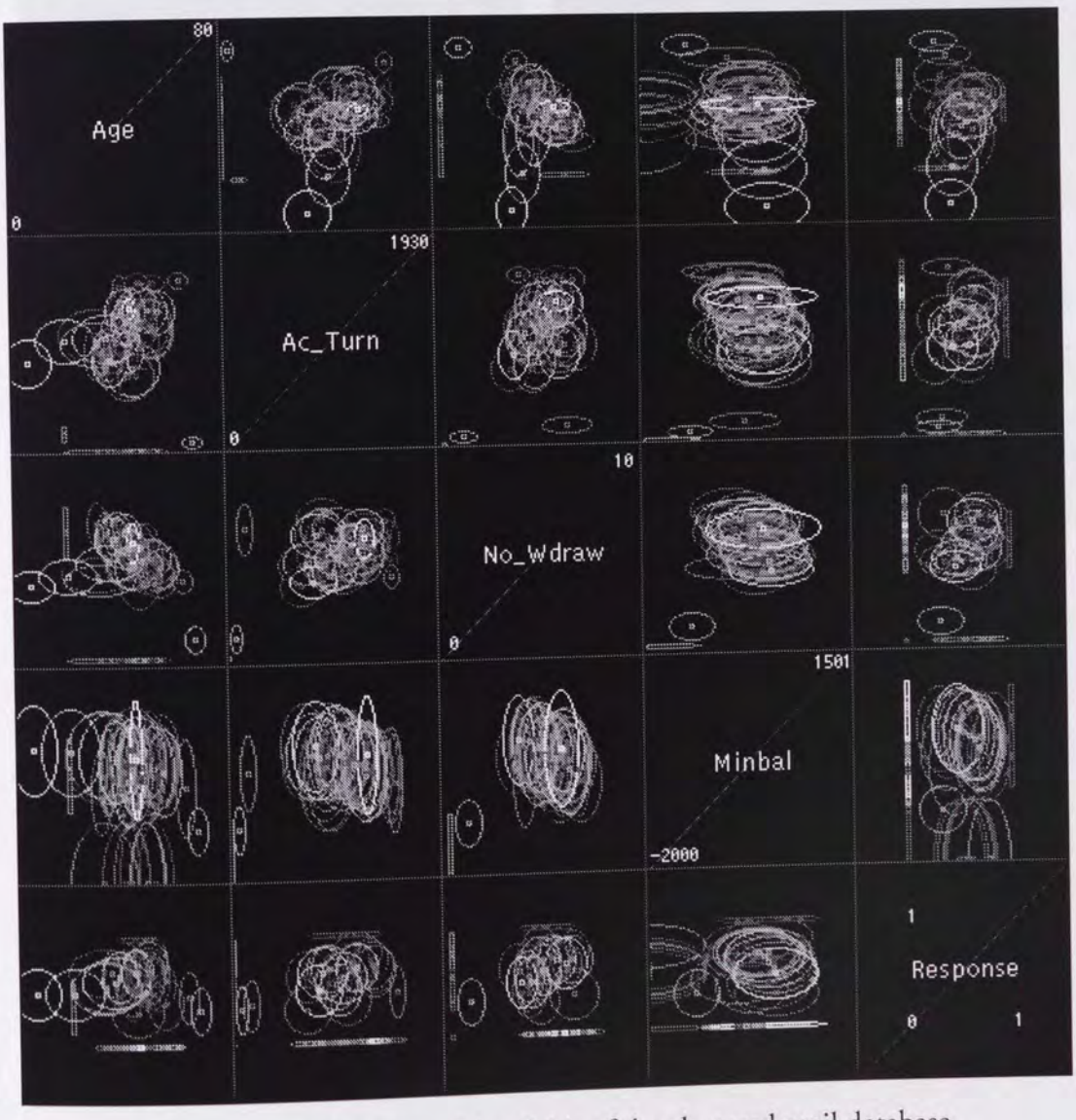
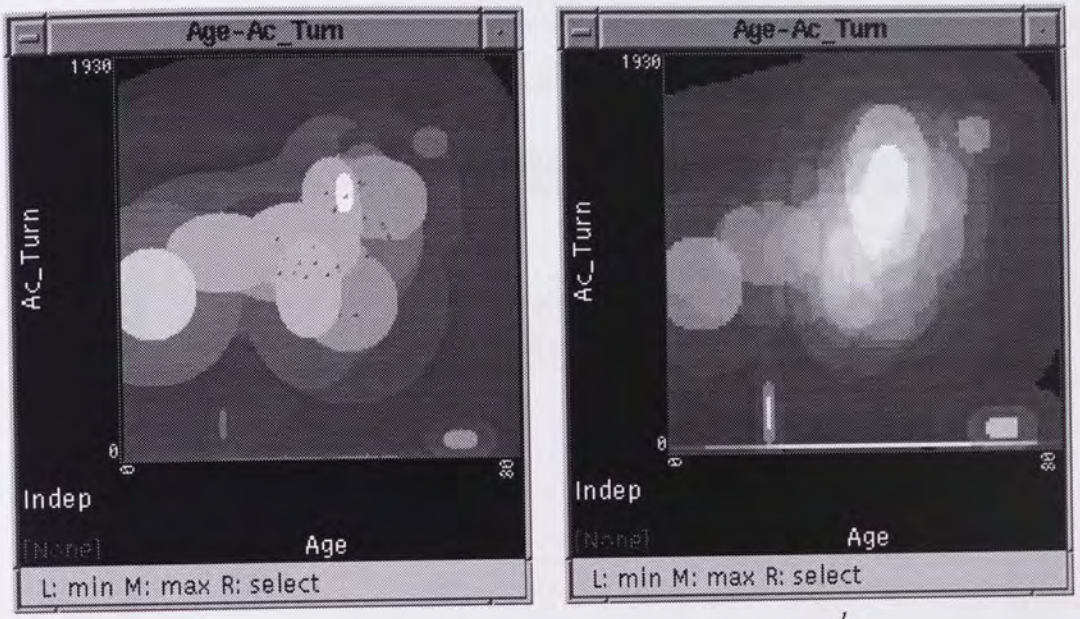


Figure 5.12 – Overview of five fields of the clustered mail database

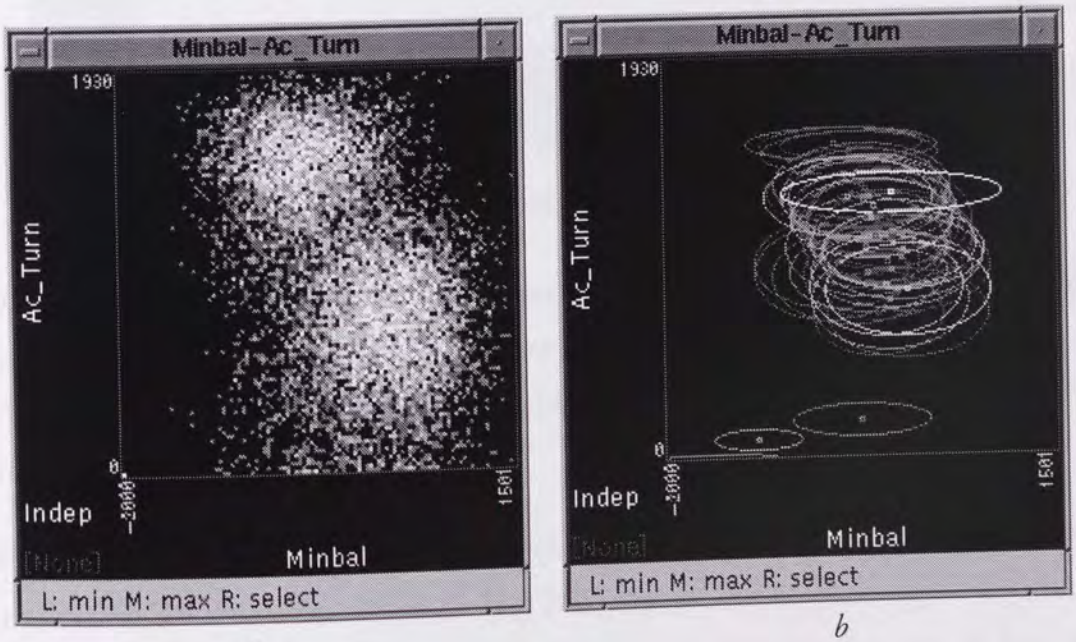
Figures 5.13 and 5.14 further demonstrate the rather poor suitability of the clusters for visualisation purposes. Figure 5.13 shows an enlargement of the same axes as used to demonstrate display techniques, using both filled ellipses and a density plot. It is very difficult to make out any of the structure of the original data seen in figure 5.3.



a *b*

Figure 5.13 – Example enlargement from the fifty clusters of the full mail database

Figure 5.14 shows a comparison between the original and clustered versions of the enlargement of `Minbal` against `Ac_Turn`. The overall shape is similar, but the clustering process has evidently not managed to model the data very well.



a *b*

Figure 5.14 – Comparison of original and clustered enlargements from the mail database

5.9.1.2 Kohonen map

A 13 × 17 Kohonen map was trained on the mail database. Figure 5.15 shows the progress of training in terms of the averaged error between the data points presented to the network and the weight vector of the winning node. The effect of decreasing neighbourhood radius is apparent, and it appears that the second phase could have been made considerably shorter without a large increase in error.

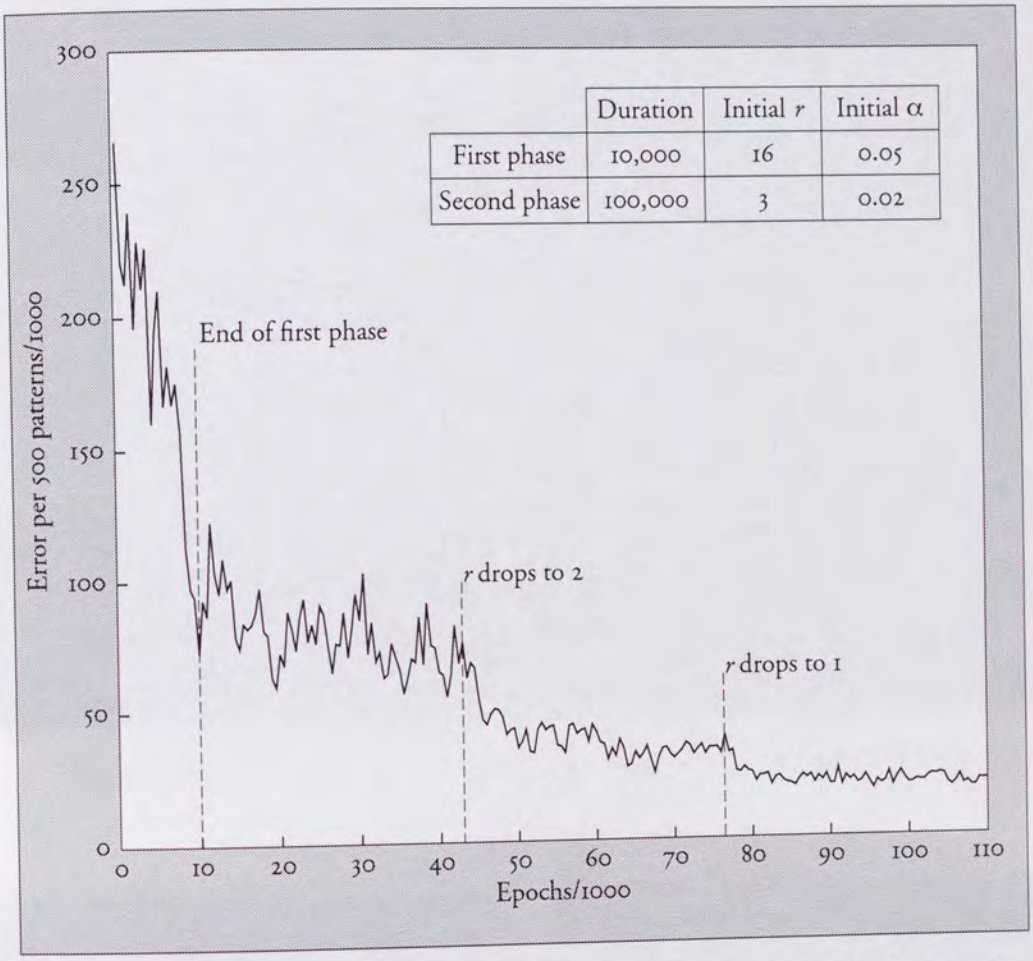


Figure 5.15 – Averaged error during training of Kohonen map on the mail database

Training completed normally, and an overview containing the augmented weight vectors was created. The overview in figure 5.15 shows six fields of the weight vectors. A lot of the structure of the original database can be seen, with a significantly faster response time.

Figure 5.16 shows the same axes of projection for the Kohonen clustering as figure 5.14 did for the mixture model clustering. Even without any indication of cluster size, it is clear that the Kohonen map has modelled the database very closely, at least in this plane.

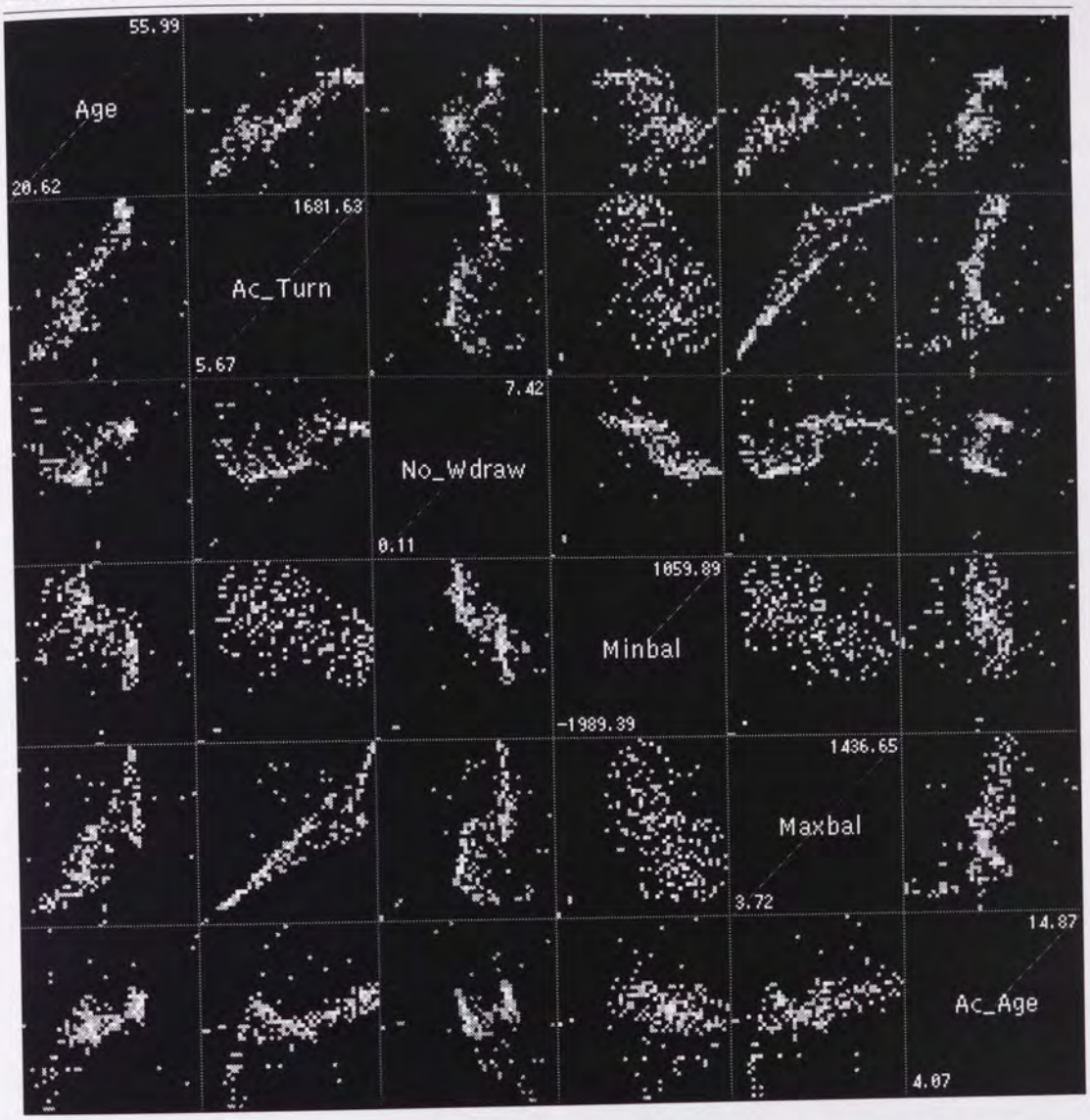
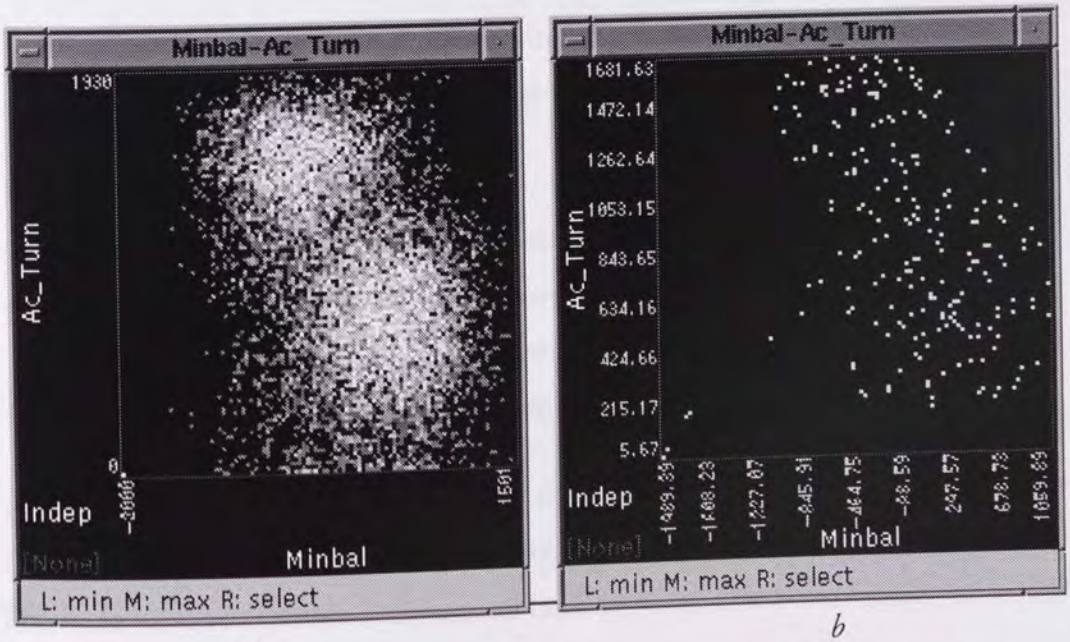


Figure 5.16 – Overview of six components of the Kohonen weight vectors from the mail database



a *b*

Figure 5.17 – Demonstration of the modelling power of the Kohonen map

Continuing, the Kohonen_den field was examined to investigate the number of data records assigned to each node. It became evident that one node was the winner for 992 records, while the rest covered a range from zero to around 100 records. As might be expected, the weights for the most 'popular' node included zero or unknown Ac_Turn, No_Wdraw, Minbal and Maxbal – so this node covers the records with missing financial information.

An enlargement window was now created, with its axes set to Kohonen_x and Kohonen_y in order to visualise the weight vectors in combination with the topography of the Kohonen map. Plate 5.1, overleaf, shows four fields being used as overlays. It should be noted that the two columns of nodes along the left edge of the map tended to have very atypical weight vectors, and very low density. They were probably located near outlying points in the data space.

Each overlay onto the Kohonen axes clearly shows that the map tends to cluster into zones with similar components, defining the customer segments:

- Plate 5.1*a* shows an area of high account turnover at the bottom of the map, an area of moderate turnover at the top, and between them a blue 'river' indicating an area of low turnover.

- Plate 5.1*b* uses Home:U as its overlay, and shows two groups of customers of unknown home status, one towards the top left of the map, the other to its right.

- Plate 5.1*c* shows a generally higher minimum balance at the top of the map than at the bottom, though there is an interesting patch of low balance towards the centre of the map.

- Plate 5.1*d*, showing the maximum balance, has a very different shape. The customers with high maximum balance are grouped at the bottom of the map, separated by a large area of low maximum balances from a group of moderate maximum balances at the top right.

The four bright blue nodes at the lower left of three of the enlargements discussed above are located at customers with unknown financial details, as previously mentioned. The node with 992 records is the second node from the left at the very bottom of the map, coordinates (1,0).

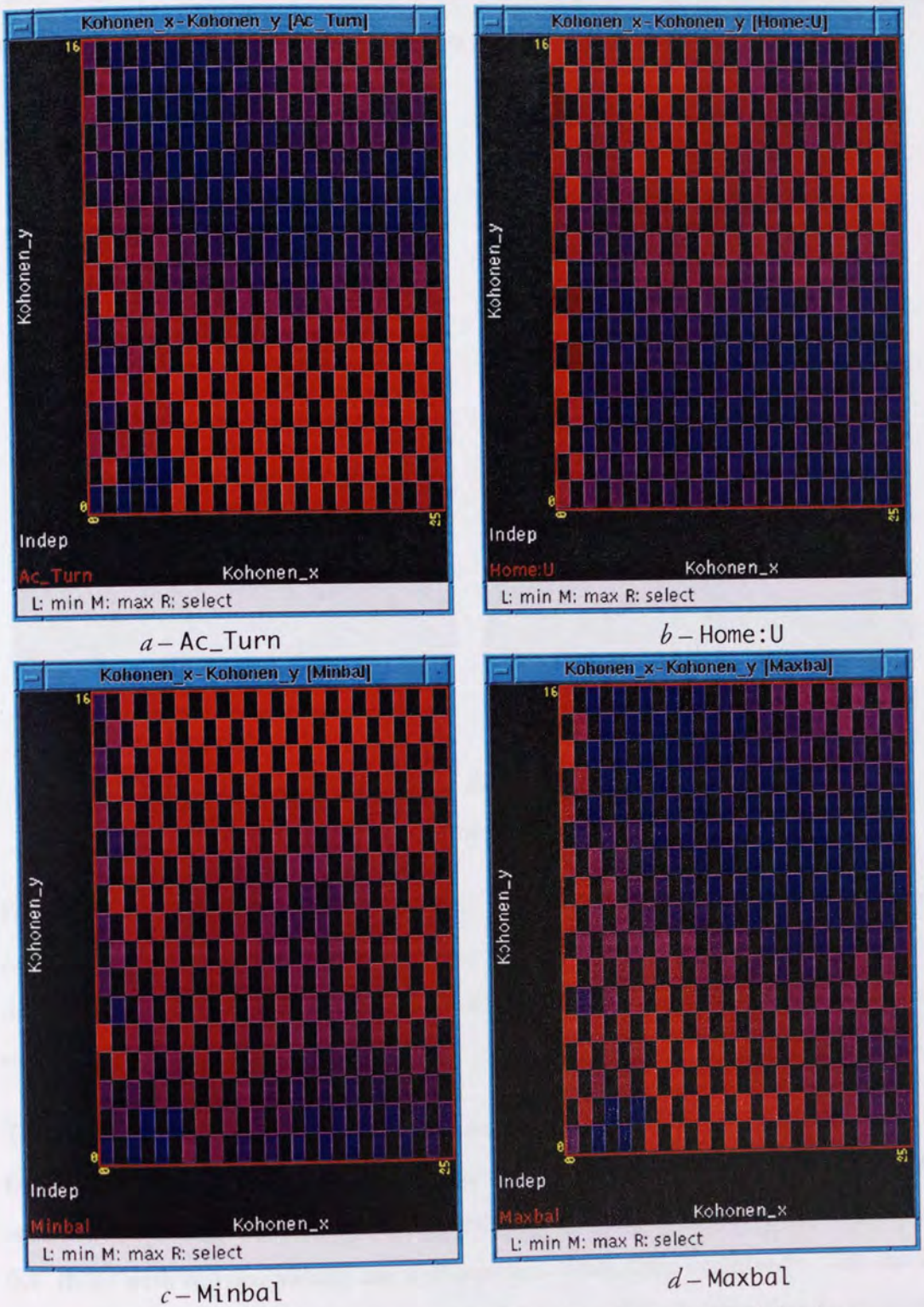
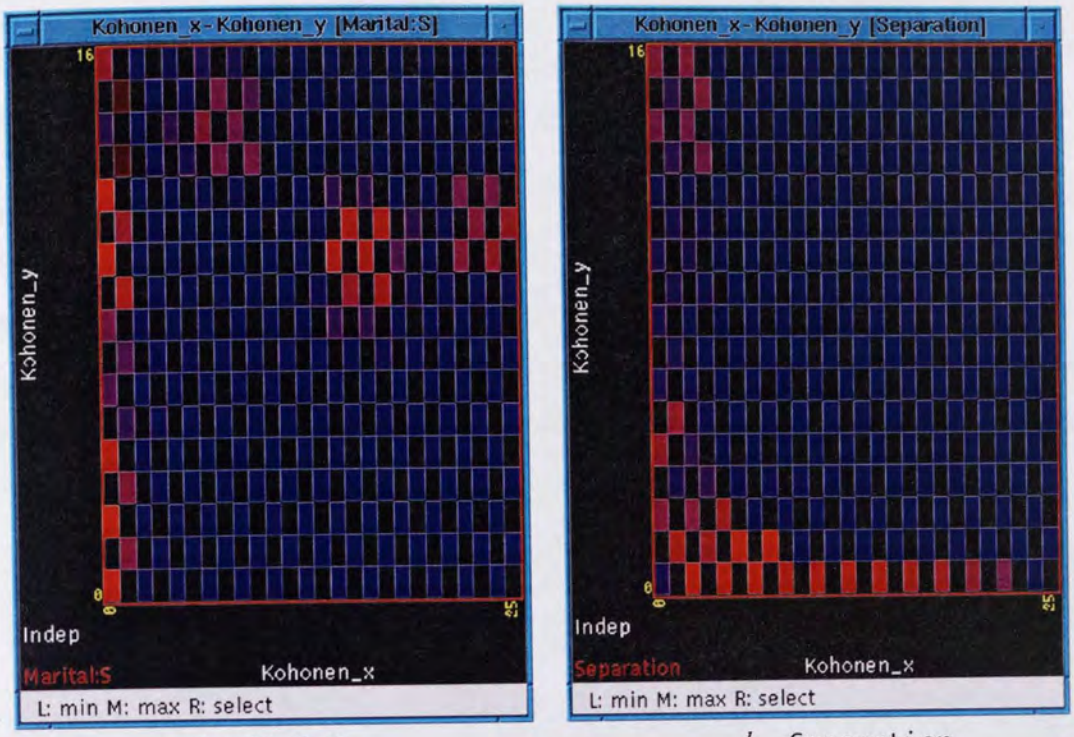


Plate 5.1 – Kohonen weight vector distributions from the mail database

By mentally combining the four enlargements, one can begin to see which types of customer are located at which locations on the map – i.e. the definitions of the segments. For example, customers mapped to the right-hand side of the map, about two-thirds of the way up, have low account turnover, a high chance of having an unknown home status, a moderately high minimum balance, and a fairly low maximum balance. Plate 5.2*a*, overleaf, shows the location of singles on the Kohonen map.

Three distinct clusters can be seen, one of which is positioned at the location under investigation. So the customers in question are fairly likely to be single.



a – `Marital:S` *b* – `Separation`

Plate 5.2 – Kohonen weight vector distribution and separation measure from the mail database

Further investigation, using more overlays, would allow many groups of customers to be located. A summary map could then be drawn, dividing the database into distinct, describable segments whose relative proximities would give an indication of their similarity to one another.

To see how closely the map followed the topography of the data, the `Separation` field was used as an overlay, resulting in the enlargement shown in plate 5.2*b*. As would be expected, the nodes covering points well away from the bulk of the data (i.e. those with missing values) are well-separated from their neighbours. The rest of the map is almost uniformly blue, indicating closely-spaced nodes spanning the occupied parts of the data space

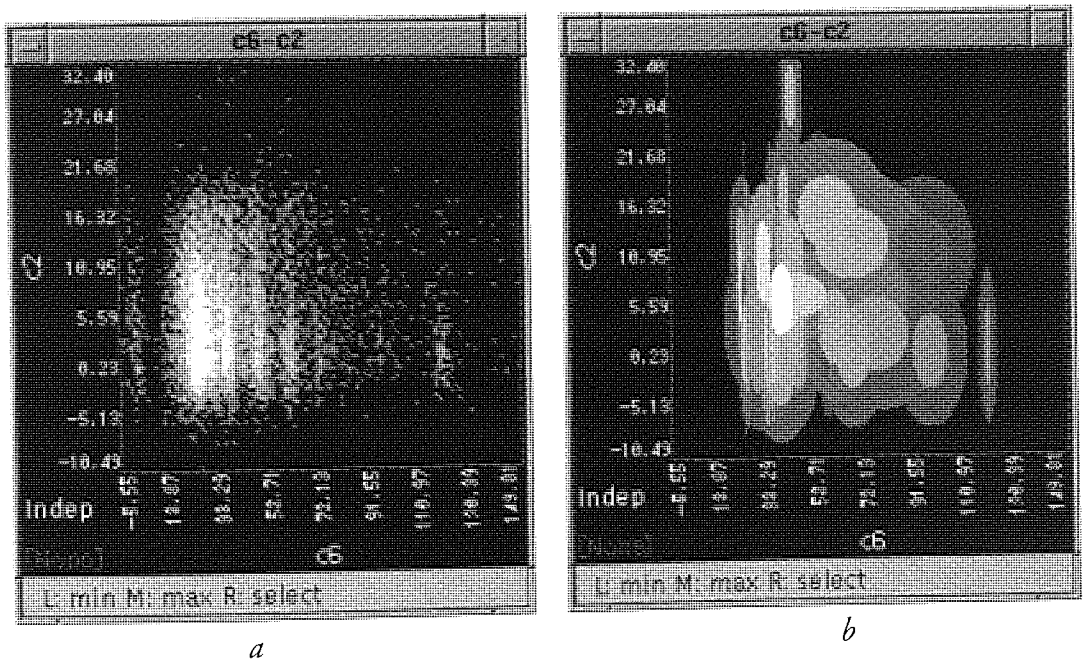
Using the system as implemented, there was no way to investigate how the response field varied across the Kohonen map. A method for doing so was developed, and will be discussed in chapter 7.

5.9.2 Finance database

5.9.2.1 Mixture model

The mixture model was unable to generate clusters for the finance database, probably due to the large number of categorical fields it contains.

In an attempt to generate some clusters, all fields except c1, c2, c3, c4, c5, c6 and Response were clipped out of the database. Even with this reduced database, there were severe problems with the clusters, none of which had a usable width on c1, c3 and c5. The fields which did have valid widths appeared to model the data fairly well. Figure 5.18 shows the clusters projected onto c6 and c2, together with the original data projected onto the same axes.



a *b*

Figure 5.18 – Comparison of original and clustered enlargements from the finance database

5.9.2.2 Kohonen map

Figure 5.19 shows the weight vectors of a 13 × 17 Kohonen map trained on the finance database. This should be compared with figure 5.20, overleaf, which shows the same fields of the original database.

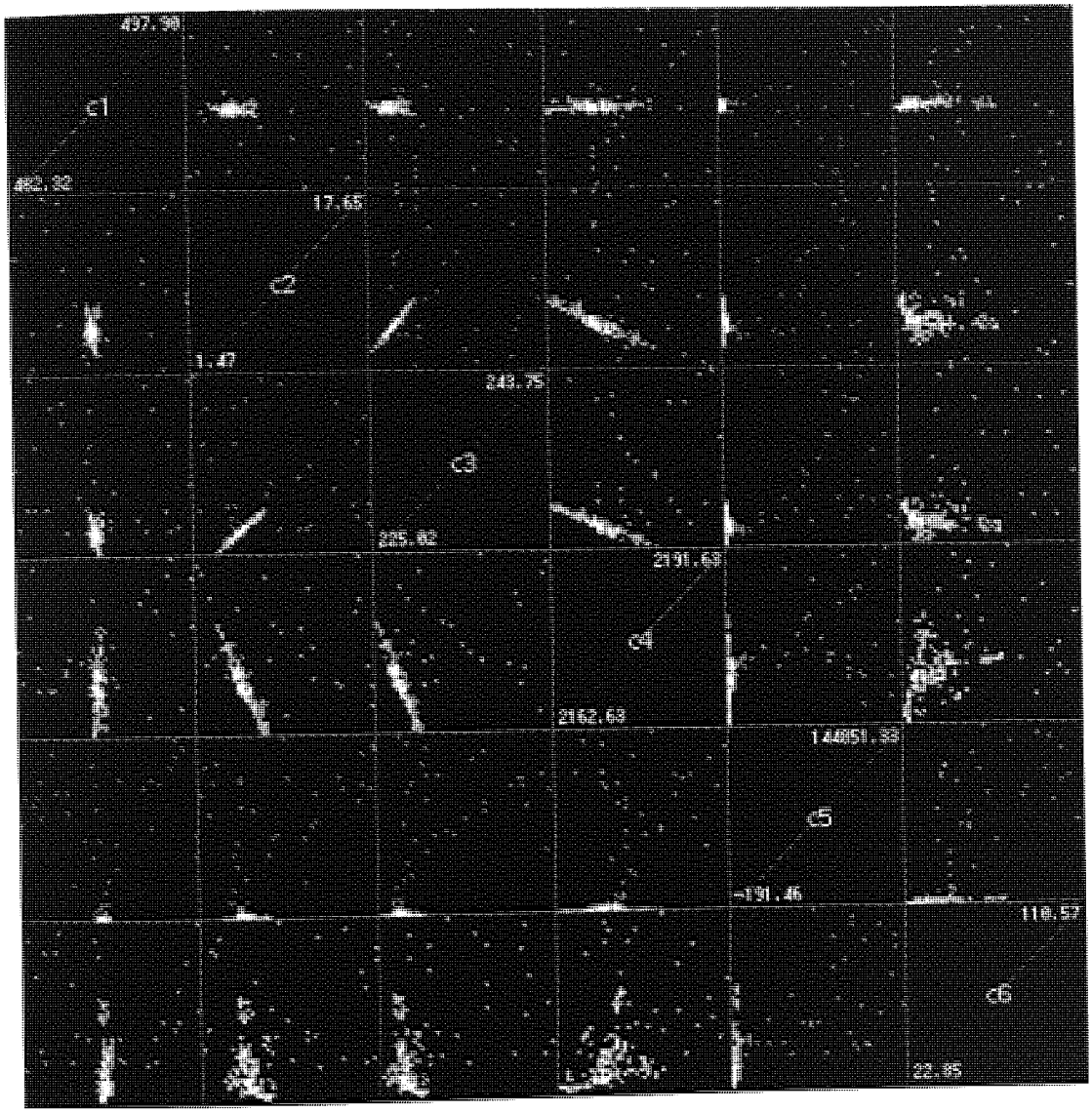


Figure 5.19 – Overview of six components of the Kohonen weight vectors from the finance database

It appears from these two figures that the Kohonen map was not as successful at mapping the finance database as it was with the mail database. Some of the overall shape is visible (e.g. in the plots of c2-c3 and c3-c4), but the fine resolution of the mail clustering is not present.

This may be due in part to the large number of categorical variables in the finance database, and in particular their expansion into numerous binary fields, which results in a total of fifty dimensions, as opposed to forty-one in the mail database

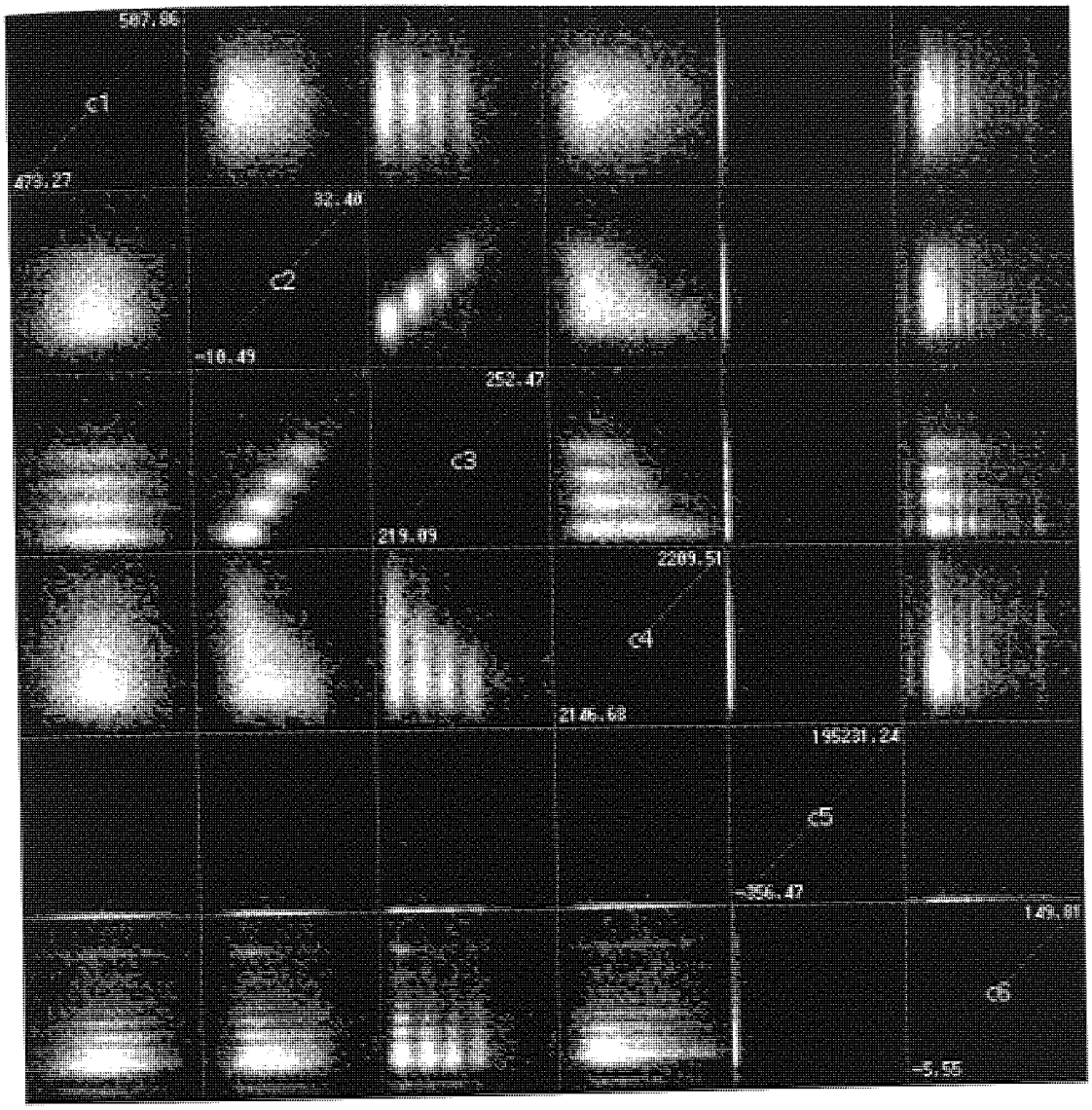


Figure 5.20 – Overview of six fields of the finance database

Even though the projections of the weight vectors seem to be unpromising, the training error definitely implied that the map had learned something about the shape of the data. To see whether any of the data's topology had been mapped, an enlargement was created in the same way as for the mail database, and various fields used as overlays.

Firstly, the `Separation` field was used, resulting in plate 5.3. This shows that the majority of the map is closely-spaced, with a few distant nodes in the lower right corner, and a large number at the left edge, presumably mapping outlying points as before.

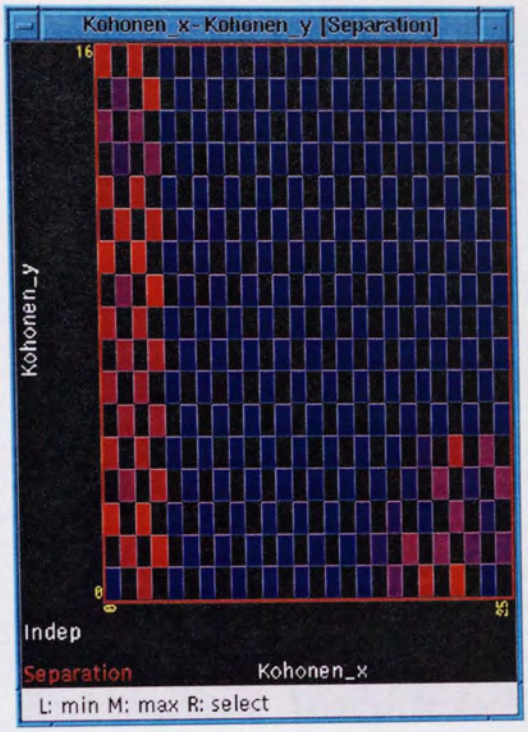


Plate 5.3 – Kohonen node separation measure from the finance database

Plate 5.4 overleaf shows some of the weight vector components overlaid on the map. As with the mail database, there are clearly-visible zones of different colours, with fairly smooth transitions between them. This would imply that the map has moulded itself fairly well around the data. If the field meanings were known, an investigation into which types of customer were located at various points on the map could be carried out.

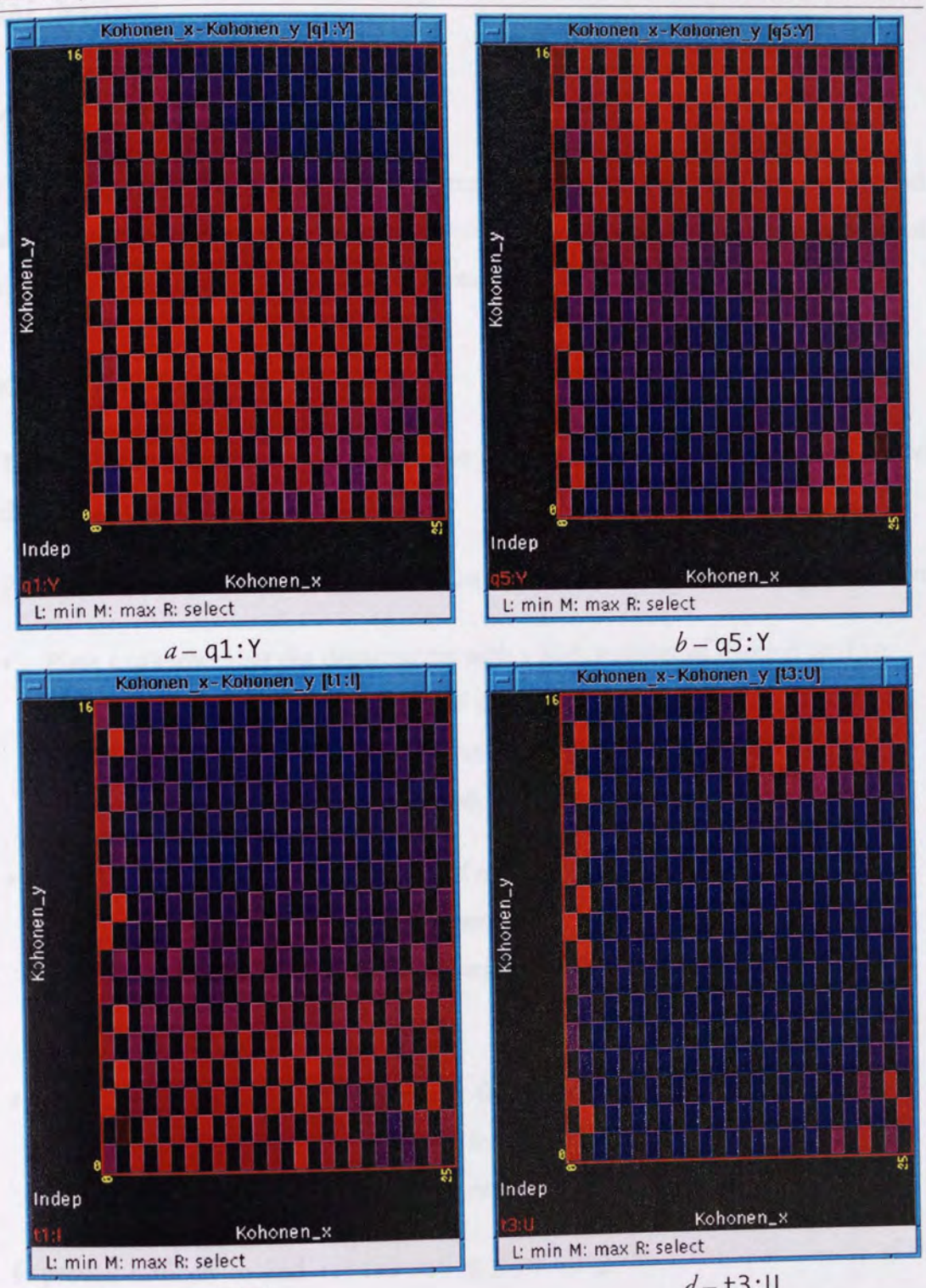


Plate 5.4 – Kohonen weight vector distributions from the finance database

5.9.3 RAE database

5.9.3.1 Mixture model

Disappointingly, the mixture model as implemented was completely unable to find clusters in the RAE database. The code failed with NAN errors almost immediately, and there was insufficient time available to investigate the cause.

5.9.3.2 Kohonen map

The Kohonen map, however, managed to generate a set of weight vectors to model the database with few problems.

Plate 5.5 shows some of the weight vector components overlaid on the map projection:

- Plate 5.5*a* shows that the departments with a high number of selected staff are clustered in the lower right corner of the map. Such departments, it will be recalled, are likely to achieve high ratings (though this information is not accessible from the weight vector plots).

- Plate 5.5*b* reveals that the distribution of `n_doctoral`, the number of doctorates, is similar to `sel_staf`, though it stretches further to the left, allowing these nodes to map departments with high numbers of doctorates but lower selected staff.

- To show that not all the overlaid fields are the same, plate 5.5*c* shows `num_grant_c`, the number of grants from UK central government. This zone is located somewhat higher than the previous two.

It proved difficult to find any interesting plots using this technique – most of the map was blue, with a small patch of more red nodes towards the lower right corner. Plate 5.5*d* helps to explain why. This shows the `Separation` variable, and shows that there are a large number of well-separated nodes at the left side of the map. Detailed analysis of these weight vectors showed that they covered departments widely different from any others, i.e. strong outliers in the database.

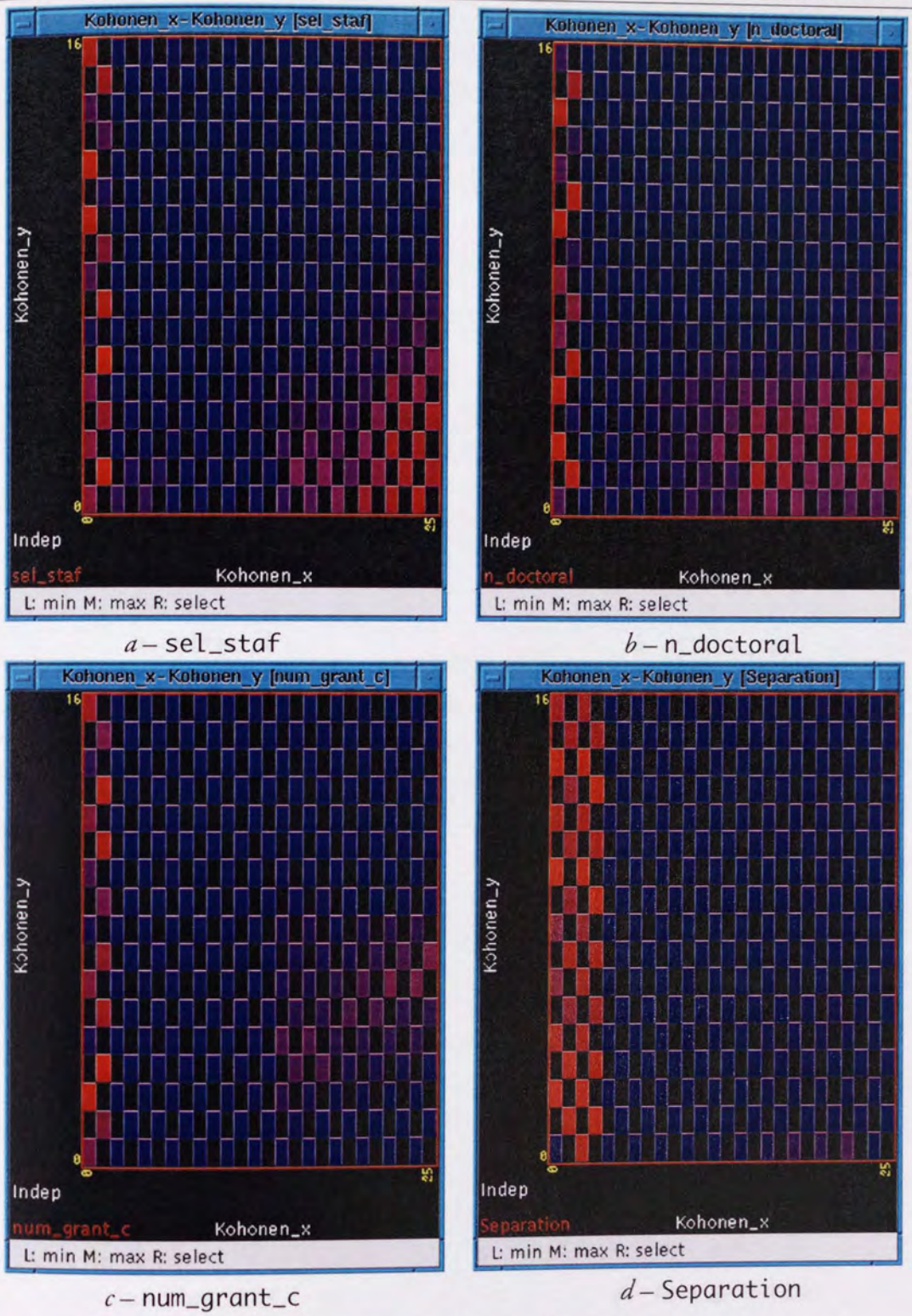


Plate 5.5 – Kohonen weight vector distributions and separation measure from the RAE database

A selection and clip operation was used to remove the left three columns of nodes from the map database. This resulted in more easily visualised clusters, since the overlay colour scale was not being scaled by the large values in the now-deleted nodes. Plate 5.6 shows four enlargements using the reduced map.

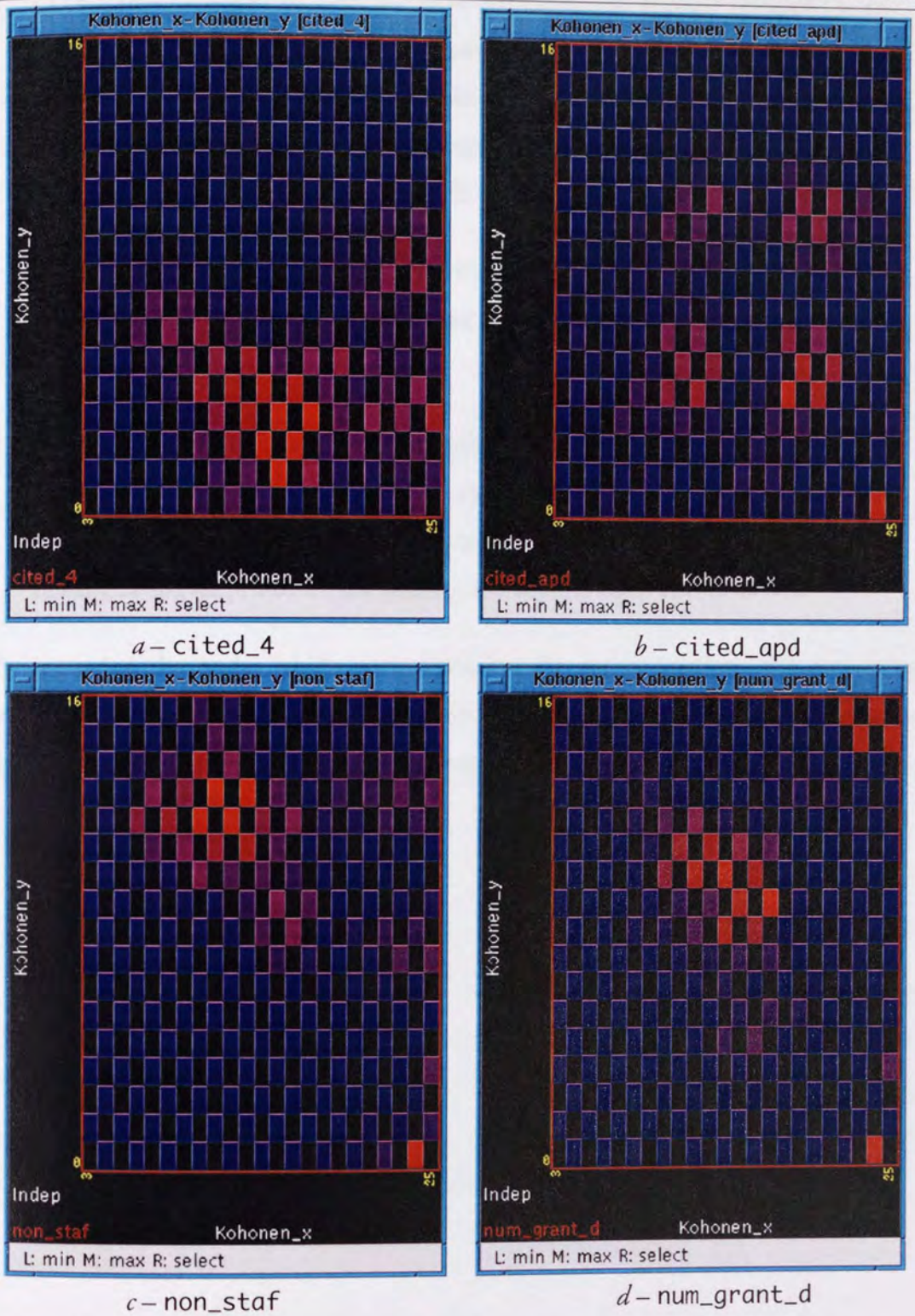


Plate 5.6 – Kohonen weight vector distributions from the RAE database (three left columns removed)

Examination of plate 5.6 allows further analysis of the clusters:

- Departments with a large number of cited refereed conference proceedings, `cited_4`, are grouped into a cluster at the lower middle of the map, as shown in plate 5.6*a*. There are also moderately-high areas on the right edge of the map.

- `cited_apd` (number of cited publications classed as applied research) is an extremely interesting vector component, as plate 5.6*b* reveals. Four small clusters of high numbers can be seen in a square pattern in the middle of the map. Also, the single node in the bottom right corner is strongly weighted towards departments having many such citations.

- Plate 5.6*c* shows that this node is also highly weighted for the number of staff not selected. This component also has significant zones in the upper half of the map.

- Finally, a component was found which places departments in the top right corner of the map. `num_grant_d`, the number of grants from UK local government, shown in plate 5.6*d*, is weighted highly in this corner, in the middle of the map, and on the node in the bottom right corner as well.

By removing the nodes which were distorting the colour overlays, the projections have become more revealing and clearly show that the Kohonen training process has mapped similar departments into small zones on the map.

5.10 Conclusions

As has been seen, the large size of the databases under consideration is a two-edged sword. It makes the use of clustering techniques desirable in order to increase processing speed, but then precludes the use of many of the more common clustering techniques.

The two 'traditional' clustering techniques did not lend themselves to automatic operation in a visualisation system. The sequential leader algorithm, though very fast, can generate vast numbers of clusters, many of which are composed of only one outlying data point. FASTCLUS provides a limit on the number of clusters, but the clusters which are formed are still merely groups of data points, with no defined structure and shape. More importantly, there are several parameters which need to be optimised before the algorithm generates suitable clusters.

The mixture model performed well on low-dimensional data, generating a set of easily-visualisable elliptical clusters, albeit at the expense of a considerable amount of processing time. However, the model proved to be unsuited for use with real high-dimensional data, especially when it contains large numbers of discrete or binary fields – though some of the failings were probably due to the particular algorithm used, rather than the mixture model concept itself.

Should a suitable set of mixture model clusters be found, they can be quickly and accurately visualised in the form of concentric ellipses. This brings the two desired results of clustering, namely speed of presentation of the cluster locations, sizes and densities, and simplification of the data's structure. The use of density plots accurately shows the density in areas covered by more than one cluster, but is too time-consuming for general use, being generally slower than displaying the original data.

A significant loss of information occurs during the clustering process. It was expected that some information would be lost, however the extent to which the shape of the original data would disappear was unexpectedly large. This is likely to be another consequence of the high dimensionality of the data.

The Kohonen map, though sharing many of the features of the other clustering techniques, is a useful tool for data visualisation. Its topography-preserving property results in an analysis technique which allows the user to draw his own conclusions about which clusters exist, and what properties are shared by the records in those clusters, allowing the segmentation of a customer database to be found and described.

In the implementation in MADEN, the use of the Kohonen_x and Kohonen_y fields allow the changes in the weight vector components to be seen in the form of the changing colour of an overlay, and the Separation field gives a clear indication of which areas of the map model smoothly-changing parts of the data, and which are stretched to cover outlying areas of the data.

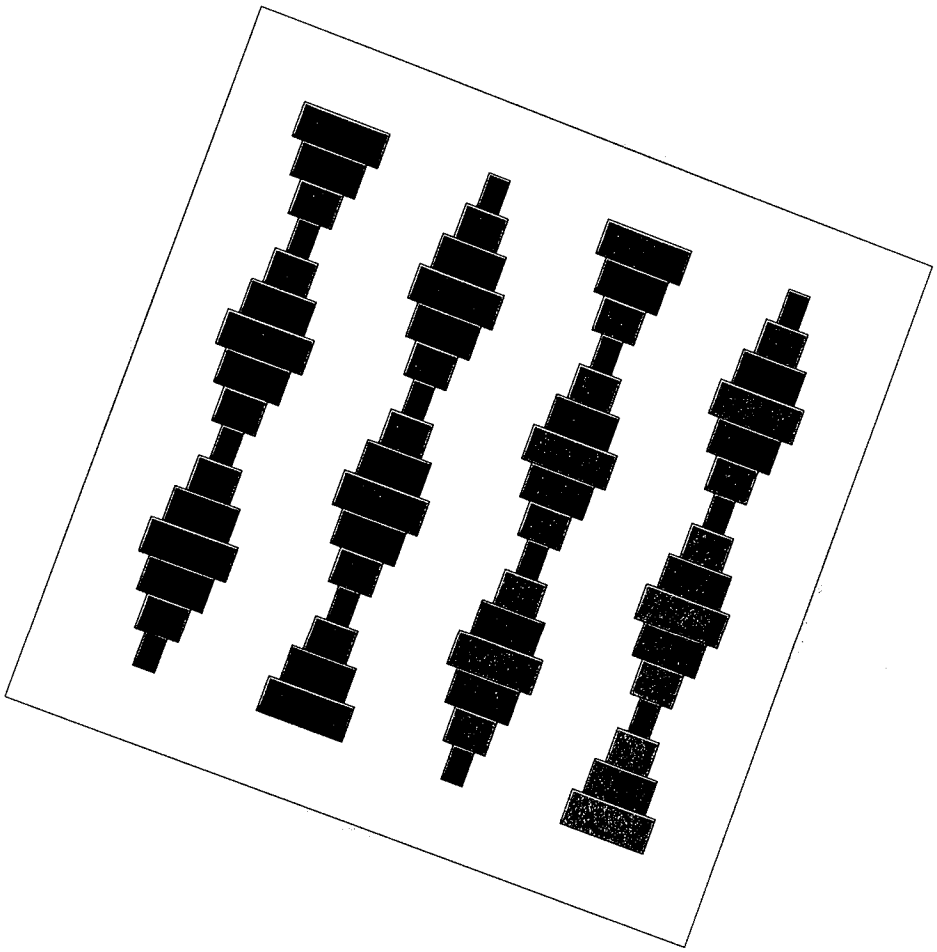
As will be seen in chapter 7, the Kohonen map offers further possibilities for data visualisation in MADEN.

5.10.1 Summary

In summary, generating clusters from the raw data is not, in general, worthwhile. The time spent in finding the clusters (if indeed any can be found) and the loss of information are too great for all but exceptionally large databases which are too large to comfortably view directly, and only then if the clustering process manages to generate a set of clusters.

However, the unconventional Kohonen technique, though taking a long time to optimise its fit to the data, offers both a data reduction (by visualising the database of the weight vectors) and also a two-dimensional map of the data which conveys far more information than the other clustering methods, particularly in the area of market segmentation.

Chapter 6



Dimensionality Reduction I: Linear Methods

It is now known to science that there are many more dimensions than the classical four. . . . This means either that the universe is more full of wonders than we can hope to understand or, more probably, that scientists make things up as they go along.

[Pratchett, 1989]

6.1 Introduction

Having seen that clustering is generally a poor way to condense the information in a database in order to aid visualisation, alternative methods of data reduction had to be sought. Clustering reduced the number of records; reducing the number of fields was the obvious alternative.

The primary aim of this approach is not to increase system speed, but rather to enable information hidden in the database to be more easily accessed by the user. With a smaller number of fields, visualisation of the entire database is likely to be feasible, as is the examination of a higher proportion of 2-D projections. If the new dimensions are suitably chosen, previously invisible structure in the database may also become apparent.

6.1.1 Trivial field reduction

A reduction in the number of fields evidently occurs every time MADEN generates an incomplete overview, and, to a greater extent, whenever an enlargement is created. Though these processes are often of use, they result in the loss of all information contained in the discarded fields.

6.1.2 Dimensionality reduction

The wider area of *dimensionality reduction* is concerned with *transforming* the data into a new multidimensional space with fewer dimensions than the original database, while retaining as much of its information as possible.

There are two immediate disadvantages of dimensionality reduction compared with the trivial field reduction method: firstly the processing of the data is often time-consuming, and secondly the axes of the new space do not have the simply-understood meanings of the original fields – e.g. it is highly unlikely that a transformed field will have a meaning like 'age'. Nevertheless, dimensionality reduction provides some immensely powerful tools for data visualisation.

6.1.3 Linear projection

The simplest class of dimensionality reduction techniques is linear projection, whereby a set of orthogonal directions through the database are identified and the data is projected onto these new axes, forming the lower-dimensional transformed database.

The remainder of this chapter assesses several linear dimensionality reduction methods which were implemented in the MADEN system. Non-linear techniques will be examined in chapter 7.

6.2 Principal Component Analysis

Principal component analysis (PCA) is a much-used technique for selecting orthogonal axes in the data space along which the *variance* of the data is maximal. The first principal component (PC) axis is the direction along which the one-dimensional projection of the data has maximum variance; in general the nth PC axis is the direction orthogonal to the first $n-1$ PC axes along which the one-dimensional projection of the data has maximum variance.

By projecting the database onto a small number of PC axes, the dimensionality of the data is reduced, but it is hoped that by retaining the subspace with the greatest variance, the most important features of the data will be retained. Geometrically, the axes of the original database are rotated and the new axes with lowest variance are discarded.

6.2.1 Procedure

It is clear that the relative scaling of the axes of the data affects the directional variances and hence the choice of PC axes, and so it is important that the data be standardised in some fashion before applying PCA. A suitable standardisation is to scale along each axis of the dataset to make the variance unity. Following this transformation, it is easy to observe that any axes with variance greater than one have a greater variance than any of the original axes.

Principal component analysis requires the covariance matrix of the data, $\mathbf{S}$, as defined by equation 6.1.

$$\mathbf{S} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T \qquad 6.1$$

$$\text{where } \bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$$

If the data is additionally standardised so that each component of its mean ($\bar{\mathbf{x}}$) is zero, then $\mathbf{S}$ becomes the average of the outer products of every data vector with itself, $\frac{1}{n}\Sigma\mathbf{x}\mathbf{x}^T$.

The PC axes are defined as the eigenvectors of **S**, determined by solving equation 6.2.

$$(\mathbf{S} - l\mathbf{I})\mathbf{a} = \mathbf{0} \qquad 6.2$$

In a p dimensional data space, p solutions to this equation are found, yielding eigenvectors $\mathbf{a}_{1\ldots p}$ with eigenvalues $l_{1\ldots p}$.

l_i is the variance of the projection onto $\mathbf{a}_i$, so the axes which maximise variance in the projection are those with the largest l_i. If the l_i and $\mathbf{a}_i$ are ordered so $l_i \geq l_j$ for all $i < j$, the linear projection onto the $k \leq p$ most significant PC axes is calculated in the normal fashion, as shown in equation 6.3.

$$y_{i=1\ldots k} = a_{i1}x_1 + \ldots + a_{ik}x_k \qquad 6.3$$

Assembling the $\mathbf{a}_i$ into the columns of a $k \times p$ matrix **A** allows the projection equation to be given in vector form:

$$\mathbf{y} = \mathbf{A}^T\mathbf{x} \qquad 6.4$$

Or, for the entire database,

$$\mathbf{Y} = \mathbf{A}^T\mathbf{X} \qquad 6.5$$

6.3 Factor Analysis

The aim of *factor analysis* is to explain the *covariance* in a dataset in terms of a small number of *latent variables* which are by their very nature unobservable [Krzanowski, 1988].

The underlying equation for the ith component of a data item $\mathbf{x}$ in terms of q latent variables $z_{1\ldots q}$ is:

$$x_i = \gamma_{i1} z_1 + \gamma_{i2} z_2 + \ldots + \gamma_{iq} z_q + e_i \qquad 6.6$$

Equation 6.6 postulates that each data record is made up of contributions from a number (q) of *common factors* (the z_i), together with a 'residual' e_i which is *specific* to that data record. The constant γ_{ij} expresses the importance of factor j in the value of field i, and is known as the *loading* of factor j on field i. The residuals e_i are random variables, with mean zero and variance ψ_i^2.

In vector notation,

$$\begin{aligned} \mathbf{x} &= \mathbf{\Gamma}^T \mathbf{z} + \mathbf{e} \\ \mathbf{X} &= \mathbf{\Gamma}^T \mathbf{Z} + \mathbf{E} \end{aligned} \qquad 6.7$$

$\mathbf{E}$ has a covariance matrix $\mathbf{\Psi}$ which is diagonal, with diagonal elements $(\psi_1^2, \psi_2^2, \ldots, \psi_p^2)$, known as the *specific variances*. ψ_i^2 determines how much of the variability of the field i is *not* attributable to the common factors, and hence how reliably each variable can be modelled by these factors.

6.3.1 Principal factor analysis

The principal components projection equation 6.4 can be used to derive a method for determining factors. Since $\mathbf{A}$ is orthogonal, $\mathbf{A}\mathbf{A}^T = \mathbf{I}$ and therefore equation 6.3 can be re-written $\mathbf{x} = \mathbf{A}\mathbf{y}$, giving equation 6.8:

$$x_{i=1\ldots p} = a_{1i} y_1 + a_{2i} y_2 + \ldots + a_{pi} y_p \qquad 6.8$$

Now if only q of the p components are used, and the remaining $p-q$ are summed to give a residual η_i,

$$x_{i=1\ldots p} = a_{1i}y_1 + a_{2i}y_2 + \ldots + a_{qi}y_q + \eta_i \qquad 6.9$$

which is identical to equation 6.6. Thus there is a close relationship between PCA and FA, which is used by the principal factor analysis method detailed below.

Principal factor analysis (PFA) [Krzanowski, 1988] is an iterated form of PCA, which does not require any assumptions to be made about the distribution of the data (unlike, for example, maximum likelihood factor analysis, which requires the data to be approximately normal). PFA uses the following simple algorithm:

1 Generate the diagonal matrix $\hat{\mathbf{\Psi}}$ containing estimates of the specific variances, using equation 6.10, where r_{ij} is an element of $\mathbf{R}$, the correlation matrix of the data.

$$\hat{\psi}_{ij} = \begin{cases} 1 - \max\left[r_{ik}\right]_{k=1\ldots p, k\neq i} & i = j \\ 0 & i \neq j \end{cases} \qquad 6.10$$

2 Conduct a principal component analysis of $\mathbf{S} - \hat{\mathbf{\Psi}}$ and write the loadings of the first q components as columns of the matrix $\hat{\mathbf{\Gamma}}$.

3 Recalculate the diagonal of $\hat{\mathbf{\Psi}}$ as the diagonal of $\mathbf{S} - \hat{\mathbf{\Gamma}}\hat{\mathbf{\Gamma}}^T$.

4 Return to step 2 with this new estimate of $\mathbf{\Psi}$.

The cycle continues until two successive pairs $\hat{\mathbf{\Gamma}}$ and $\hat{\mathbf{\Psi}}$ are identical to within a small tolerance.

6.3.1.1 Explanation

Once a set of factors has been determined using PFA, equation 6.11 uses the eigenvalues l from the PCA in step 2 to give an indication of e, the proportion of the covariance which is *explained* by the factors.

$$e = \sum_{i=1}^{q} l_i \Big/ \sum_{i=1}^{p} l_i \qquad 6.11$$

6.3.2 Factor rotation

A common action following the determination of a set of factors is the assignment of a description of each factor's real meaning – for example in the investigation of the RAE database in section 6.8.3.1, the most influential factor will be seen to be 'size of department'. With other factors, and other databases, the meaning may not be as obvious. Factor rotation is an attempt to make the interpretation of axes easier, by rotating the factor axes in the low-dimensional factor space.

A common and easily-implemented method of factor rotation is *varimax* rotation [Kaiser, 1958]. This process rotates the axes iteratively to maximise an index V defined by equation 6.12. The larger the value of V, the 'simpler' the factor axes are. In practice, this 'simplification' tends to concentrate the components along each axis into one factor, and reduce the corresponding components in the other factors. This will be demonstrated in section 6.8.1.2.

$$V = \frac{1}{p^2}\sum_{j=1}^{q}\left\{ p\sum_{i=1}^{p}\beta_{ij}^4 - \left(\sum_{i=1}^{p}\beta_{ij}^2\right)^2 \right\} \qquad 6.12$$

where $\beta_{ij} = \gamma_{ij} \sqrt{\sum_{i=1}^{p}\gamma_{ij}^2}$

6.3.3 Choice of number of factors

The factors obtained by factor analysis depend on the number of factors requested: the first axis of a two-factor solution will not necessarily be the same as the axis returned by a one-factor solution. The choice of number of factors is not easy to make automatically, so it is left up to the user. There is a theoretical maximum number of factors which can be determined, which is given by equation 6.13.

$$q_{\max} = p - \frac{\sqrt{8p+1}-1}{2} \qquad 6.13$$

Importantly, the explaining power of the factors, e, does not always increase with increasing q – occasionally e will drop slightly between q and $q+1$ factors.

6.3.4 Projection

The naïve method to accomplish dimensional reduction using the common factors would be to multiply the data matrix by $\hat{\mathbf{\Gamma}}$ in the same manner as for PCA. While this does produce reasonable results, there are more sophisticated methods for estimating the factor scores $z_{1\ldots q}$ for each data record. They both involve the creation of a matrix $\hat{\mathbf{A}}$ which is then used to project the data in the normal way ($\hat{\mathbf{Z}} = \hat{\mathbf{A}}^T\mathbf{X}$).

$$\hat{\mathbf{A}}_1^T = \hat{\mathbf{\Gamma}}^T(\hat{\mathbf{\Gamma}}\hat{\mathbf{\Gamma}}^T + \hat{\mathbf{\Psi}})^{-1} \qquad 6.14$$

$$\hat{\mathbf{A}}_2^T = (\hat{\mathbf{\Gamma}}^T\hat{\mathbf{\Psi}}^{-1}\hat{\mathbf{\Gamma}})^{-1}\hat{\mathbf{\Gamma}}^T\hat{\mathbf{\Psi}} \qquad 6.15$$

In practice, it was found that these equations gave very similar results. Equation 6.14 was chosen for use, as it is faster to compute – two multiplications, one addition and an inversion, compared with four multiplications and two inversions for equation 6.15.

6.4 Directed Principal Component Analysis

When tested, it was found that PFA gave identical, or near-identical, axes to PCA. This is because the PCA is carried out on normalised data, which makes the covariance matrix **S** equal to the correlation matrix **R**.

In an attempt to direct the PCA towards axes which may be of more interest to the user, the rows and columns of **S** were multiplied by the correlation of the row or column field with the response field, generating a new scaled covariance matrix $\mathbf{S}^*$ with elements s^*_{ij} as defined by equation 6.16, where r_k is the correlation of the kth field with the response field.

$$s^*_{ij} = s_{ij} r_i r_j \qquad 6.16$$

The eigen system of $\mathbf{S}^*$ is then found and the process continues as for standard PCA.

The scaling has the effect of increasing the importance of fields which are more correlated with the response, and decreasing that of less correlated fields.

This novel technique was named 'directed principal component analysis', or DPCA.

6.5 Projection Pursuit

6.5.1 Introduction

The aim of DPCA is to direct the search for projection axes towards those which reveal something 'interesting' about the data – in this case correlations with the response field. In general, techniques which attempt this sort of search are known as projection pursuit [Friedman & Tukey, 1974; Huber, 1985; Jones & Sibson, 1987].

Specifically, projection pursuit (PP) is a generic term for any method which seeks to choose a set of orthogonal axes for linear projection in order to maximise some measurable quantity. As such, all the techniques already discussed in this chapter are simple forms of PP.

Two-dimensional PP algorithms which automatically seek out interesting projections of the data are of particular interest for visualisation.

6.5.2 Algorithm

The process is iterative:

1 Pick two initial axes

2 Project the data onto the chosen axes

3 Calculate an index which measures how 'interesting' the projection is, and the derivative of this index with respect to the axis directions

4 Using a standard optimisation technique, adjust the axes according to the derivative and return to step 2

6.5.3 Projection indices

The public domain program visualisation program *xgobi* includes PP, and allows free use of its code. Several of the projection indices used by *xgobi* were converted for use in MADEN, but the more advanced ones (Hermite, natural Hermite, Legendre, entropy and Friedman-Tukey) proved far too slow in operation when using large databases. Three of the simpler indices (holes, central mass and skew) did prove useable, but as

no documentation for these was available, the descriptions below had to be reverse-engineered from the source code.

6.5.3.1 Common elements

For the purposes of these descriptions, let x_i be the projection of the ith data point onto the first (i.e. x) PP axis, and y_i the second. The projections can then be transformed by a gaussian function, giving h_i^x and h_i^y as shown in equation 6.17: h_i^x and h_i^y will be large when the projected point lies near the origin.

$$\begin{aligned} h_i^x &= \mathrm{e}^{-x_i^2/2} \\ h_i^y &= \mathrm{e}^{-y_i^2/2} \end{aligned} \qquad 6.17$$

The means of h_i^x and h_i^y are then found, as in equation 6.18:

$$\begin{aligned} \mu^x &= \frac{1}{n}\sum_{i=1}^{n} h_i^x \\ \mu^y &= \frac{1}{n}\sum_{i=1}^{n} h_i^y \end{aligned} \qquad 6.18$$

6.5.3.2 Holes and central mass indices

The 'holes' and 'central mass' indices are closely related – in fact they are inverses: a projection with maximum I_{holes} has minimum $I_{centralmass}$, and vice versa. Both require the use of a combined quantity ω as defined in equation 6.19, which is a measure of the covariance of the transformed data. ω will be minimal when the projected data is clustered at the origin.

$$\omega = \mu^x \mu^y + \frac{1}{n}\sum_{i=1}^{n} (h_i^x - \mu^x)(h_i^y - \mu^y) \qquad 6.19$$

Having found ω, the two indices are given by equations 6.20 and 6.21. $I_{centralmass}$ will be large when the projected data is clustered at the origin; I_{holes} will be large when there is a 'hole' at the origin with no data.

$$I_{holes} = \frac{1-\omega}{1-e^{-1}} \qquad 6.20$$

$$I_{centralmass} = \frac{\omega - e^{-1}}{1-e^{-1}} \qquad 6.21$$

6.5.3.3 Skew index

The 'skew' index requires two more 'means', λ^x and λ^y, as defined by equation 6.22:

$$\begin{aligned} \lambda^x &= \frac{1}{n}\sum_{i=1}^{n} x_i e^{-x_i^2/2} = \frac{1}{n}\sum_{i=1}^{n} x_i h_i^x \\ \lambda^y &= \frac{1}{n}\sum_{i=1}^{n} y_i e^{-y_i^2/2} = \frac{1}{n}\sum_{i=1}^{n} y_i h_i^y \end{aligned} \qquad 6.22$$

Next, two quantities ω^x and ω^y are defined as in equation 6.23:

$$\begin{aligned} \omega^x &= \mu^x\lambda^y + \frac{1}{n}\sum_{i=1}^{n}(h_i^x - \mu^x)(y_i h_i^y - \lambda^y) \\ \omega^y &= \lambda^x\mu^y + \frac{1}{n}\sum_{i=1}^{n}(x_i h_i^x - \lambda^x)(h_i^y - \mu^y) \end{aligned} \qquad 6.23$$

The index I_{skew} is then defined as the sum of the squares of these two quantities, as shown in equation 6.24:

$$I_{skew} = \left(\omega^x\right)^2 + \left(\omega^y\right)^2 \qquad 6.24$$

6.5.4 Implementation

The three PP indices were implemented as subclasses of a generic C++ PP object, which was made suitable for use by the standard conjugate gradient (CG) minimisation routine [Press *et al*, 1992] contained in Mike Tipping's library code – which meant negating the indices, in order that minimisation would result in maximising the index itself.

Optimisation is then a simple matter of constructing the CG minimiser, passing it the PP object and letting it run.

6.6 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) differs from the previous methods (except DPCA) in that it takes group information into account when selecting a set of axes. The aim is to find the directions which most clearly reveal *differences* between the groups. The results of projecting the data onto these directions are known as the *canonical variates.* Specifically, the canonical variates are the projections which give the largest ratio of variance between groups to variance within groups.

6.6.1 Procedure

In the following discussion, the data is classified into g separate groups with n_i individuals in the ith group. The j th member of the i th group is denoted by $\mathbf{x}_{ij}$ and the mean of the members of the i th group by $\bar{\mathbf{x}}_i$.

The *between-groups sum-of-squares and -products matrix* $\mathbf{B}_0$ is then defined as the weighted sum of the covariances of the mean of each group:

$$\mathbf{B}_0 = \sum_{i=1}^{g} n_i(\bar{\mathbf{x}}_i - \bar{\mathbf{x}})(\bar{\mathbf{x}}_i - \bar{\mathbf{x}})^T \qquad 6.25$$

and the *within-groups sum-of-squares and -products matrix* $\mathbf{W}_0$ as the sum of the covariances within each group:

$$\mathbf{W}_0 = \sum_{i=1}^{g}\sum_{j=1}^{n_i}(\mathbf{x}_{ij} - \bar{\mathbf{x}}_i)(\mathbf{x}_{ij} - \bar{\mathbf{x}}_i)^T \qquad 6.26$$

Then, dividing by the appropriate degrees of freedom, the *between-groups covariance matrix* $\mathbf{B}$ and the *within-groups covariance matrix* $\mathbf{W}$ are given by equation 6.27.

$$\mathbf{B} = \frac{1}{g-1}\mathbf{B}_0$$
$$\mathbf{W} = \frac{1}{n-g}\mathbf{W}_0 \qquad 6.27$$

For a given direction **a**, the ratio which reveals how separated the groups will be is equation 6.28:

$$F = \frac{\mathbf{a}^T\mathbf{B}\mathbf{a}}{\mathbf{a}^T\mathbf{W}\mathbf{a}} \qquad 6.28$$

At the maximum value of F, the derivative of F with respect to **a** will be zero. Thus,

$$\frac{dF}{d\mathbf{a}} = \mathbf{B}\mathbf{a} - \left(\frac{\mathbf{a}^T\mathbf{B}\mathbf{a}}{\mathbf{a}^T\mathbf{W}\mathbf{a}}\right)\mathbf{W}\mathbf{a} = \mathbf{0} \qquad 6.29$$

The bracketed term in equation 6.29 is F, which is a constant at this maximum point. Denoting this constant by l yields equation 6.30:

$$\mathbf{B}\mathbf{a} - l\mathbf{W}\mathbf{a} = \mathbf{0} \qquad 6.30$$

which can be rewritten in the form of an eigen problem:

$$(\mathbf{W}^{-1}\mathbf{B} - l\mathbf{I})\mathbf{a} = \mathbf{0} \qquad 6.31$$

The solution of this equation results in p eigenvectors $\mathbf{a}_i$ with associated eigenvalues l_i. However, it can be shown that at least $(p - g + 1)$ of the eigenvalues will be zero, and that there will generally be $s = \min(p, g - 1)$ non-zero eigenvalues. Each eigenvector is a linear discriminant axis, and the corresponding eigenvalue is the ratio F, i.e. the amount of separation between groups achieved by projecting onto the discriminant axis.

The projection of the data onto the LDAs thus found creates the canonical variates **Y**. If the eigenvectors are arranged as the columns of a matrix **A**, then $\mathbf{Y} = \mathbf{A}^T\mathbf{X}$.

In the case where $g = 2$, visualisation of the LDA is difficult because there is only one canonical variate. To overcome this, the matrix of total covariance $\mathbf{B} + \mathbf{W}$ is used in place of **B** in equation 6.31. This generates a full set of p eigenvectors, only the first of which has any significance for separating the groups. The second eigenvector (which always has eigenvalue unity) is used to generate a 'pseudo canonical variate' for visualisation purposes.

6.7 Implementation

6.7.1 Initiation

The linear dimensionality reduction techniques are initiated by the user by making a selection of 'PCA', 'PFA', 'DPDA', 'PPholes', 'PPcentralmass', 'PPskew' or 'LDA' from the 'Linear' menu at the top of the overview window.

6.7.2 Pre-processing

Before attempting to find any projection axes in the data, the database has to be prepared. Three operations are carried out:

- Serial number fields are 'removed', by setting all records to have to the mean value for these fields. The fields are not actually removed from the database, as this would have required tedious processing when generating the projections.
- Every (non-serial) field is normalised to zero mean, unit variance.
- Unless the requested operation is LDA, categorical fields are expanded into multiple fields, as in the choice of overlay – e.g. in the mail database the single `Acorn` field becomes eleven separate binary fields `Acorn:A` to `Acorn:L`. Category expansion is not performed for LDA as the resulting highly-correlated binary fields made the eigen problem solving routine unstable.

6.7.3 Choice of number of factors

For factor analysis, the number of factors to be determined has to be specified. This is achieved by presenting the user with a window divided vertically into one section for each possible number of factors, from one to $q_{\max}$ as determined by equation 6.13. The user clicks the left mouse button on the number of factors desired.

6.7.4 Choice of active PP fields

The PP routines (from *xgobi*) are capable of optimising the projection axes by changing only certain 'active' fields of the axis directions. It was decided that the simplest way to implement this was to use the list of fields which are currently displayed in the overview window. Thus if the user wishes to find an optimal 2-D projection taken from a particular four-space, the axis choice window should be used to shown only the four fields in question in the overview before requesting PP. The initial axis directions are set to be random unit-length vectors in the space of the active fields, with other field components set to zero.

6.7.5 Display of axes

Once a set of axes has been determined, it is presented graphically in an 'axis choice' window. Each field (possibly with the categorical fields expanded) of the database is listed down the left side of the window, and each axis (e.g. each PC axis) is shown as a vertical strip whose width changes with each field. The width of each portion of the strip corresponds to the magnitude of the component of the axis in the direction of the relevant field. To indicate direction, positive components are shown in red, negative ones in blue.

At the top of the window, the axes are numbered, and a horizontal green strip displays the relative 'significance' of each axis. For PCA, DPCA and LDA, the strip tapers to the right since the axes are ordered by the variance of the projection or the ratio F as appropriate. For PFA a single bar shows the explaining power of the set of axes, *e.* The numeric values of these significances are also shown. Beneath the strip, a line of numbers identify the axis strips below.

Plate 6.1 overleaf shows the axis choice window displaying the complete set of PCA axes for the mail database. Clearly, many of the axes are of such small significance that they can be discarded – indeed, this is the whole point of performing PCA. Clicking the middle mouse button on an axis strip discards all the axes to the right of the chosen one and redraws the window with the reduced set of axes, which are now wider, and hence clearer.

6.7.6 Factor post-processing

In the case of PFA axes, two operations may be carried out before projection, initiated by sequential clicks of the right mouse button in the axis choice window. The first click performs varimax rotation, the second simply reverses any axes whose components have a negative mean. This 'flipping' operation aims to further aid axis interpretation. A third click returns to the factor choice display described in section 6.7.3.

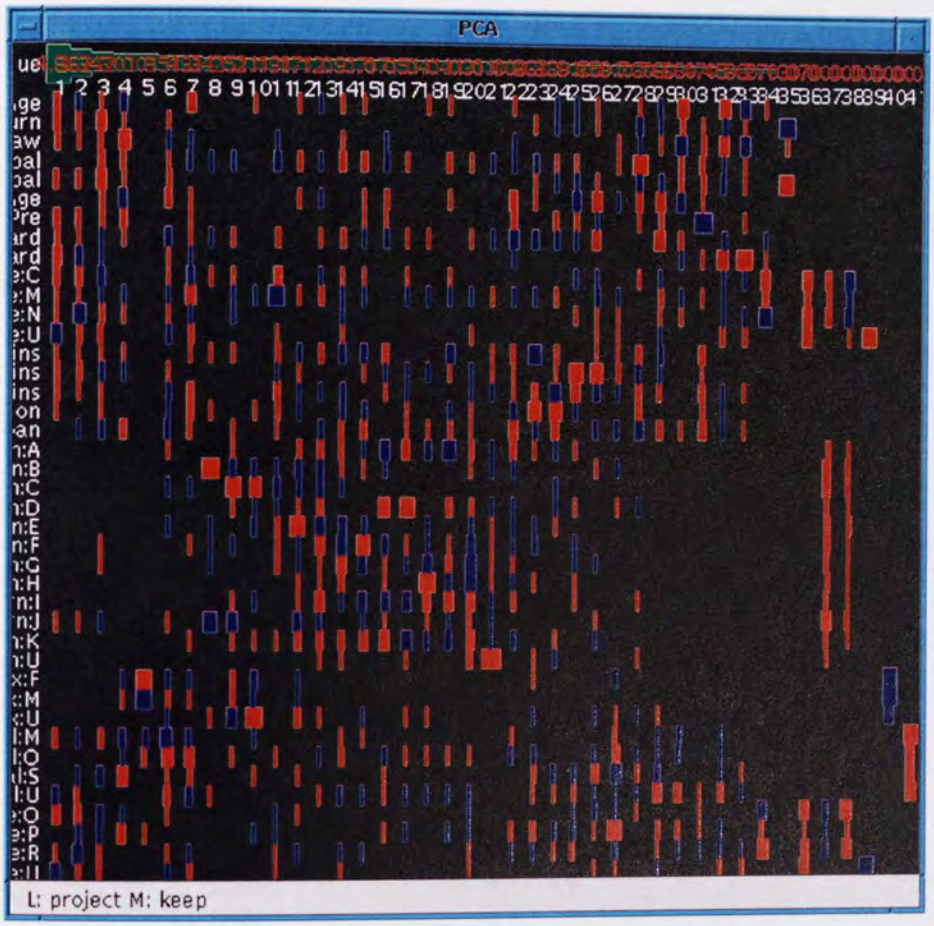


Plate 6.1 – Axis choice window following PCA on the mail database

6.7.7 Projection

Clicking the left mouse button on an axis strip generates a projection onto the axes up to and including the chosen one. A new overview window is created to display the transformed database.

The decision was taken to copy the original fields into the new database, rather than to discard them, because the user might wish to examine relationships between original and transformed fields – particularly the response field. The database can of course be clipped to include only the required fields at a later time.

6.8 Use with Real Data

6.8.1 Mail database

6.8.1.1 Principal component analysis

Figure 6.1 shows the first six PC axes of the mail database.

For clarity, all figures showing axis choice windows have been inverted. The darker shade is red (i.e. positive components), the lighter shade is blue (i.e. negative components).

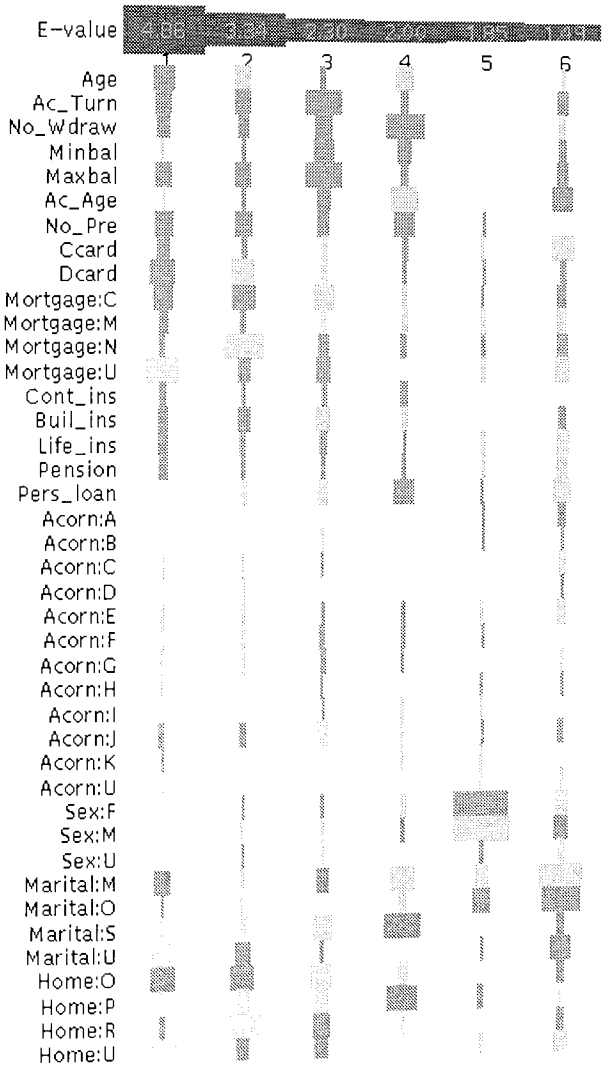


Figure 6.1 – First six principal components of the mail database

An initial, visual, analysis of the implications of these axes led to the findings below. It must be remembered that each axis creates a scale along which every customer is measured: the descriptions given apply to one end of this scale; the opposite description applies to the other end.

- The most significant component emphasises customers who tend to be older, married home owners with high account turnover and maximum balance, both credit and debit cards, all types of insurance and a pension. Customers with unknown mortgage, marital and home status are placed strongly at the other end of the scale.

- The second component is best understood by mentally inverting it. It now emphasises older customers who rent their houses, have a debit card and a low maximum balance.

- The third component appears to be a 'wealth index'. Maximum and minimum balances are high, as is account turnover, though age is insignificant. Interestingly, it is biased against singles, and customers who own their homes or live with their parents.

- The fourth component clearly picks out the younger single customers who live with their parents. They tend to have young accounts, a low account balance, often have a personal loan and make many withdrawals.

- The fifth component separates men and women; the sixth separates the married customers.

Figure 6.2 overleaf shows an overview of the mail database projected onto the first six PC axes. Examining the figure shows many interesting clusters of customers, which would bear close scrutiny by a marketing manager. For example, the plot of PC_3, the 'wealth index', against PC_4, the 'youth detector' shows three distinct clusters: a large one with moderate to high wealth towards the 'not so young' end of PC_4, and two smaller clusters with low wealth, one at each end of the 'youth' spectrum – presumably one is the poor students, the other the poor OAPS.

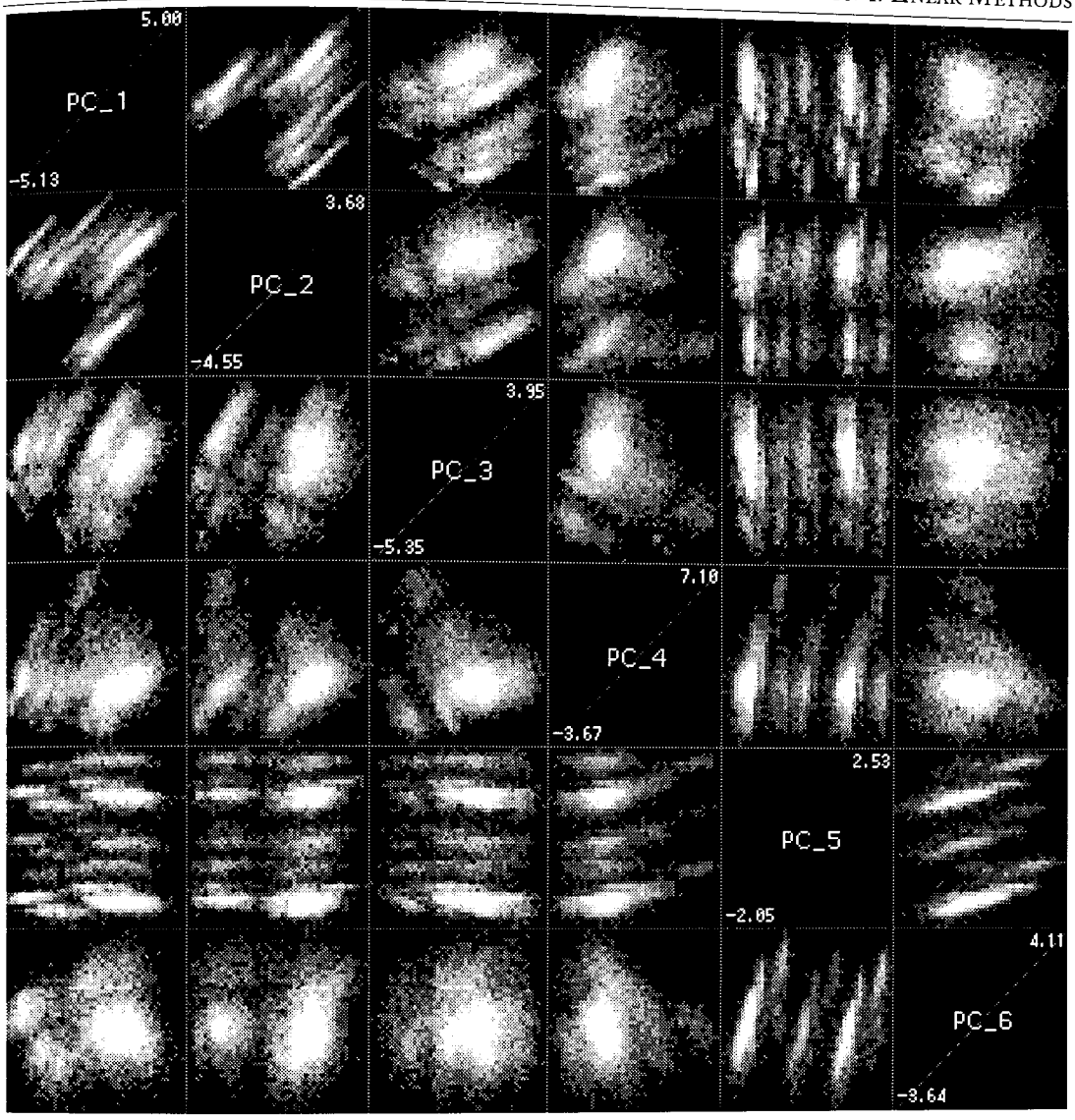


Figure 6.2 – Projection of the mail database onto its first six principal components

6.8.1.2 Factor analysis

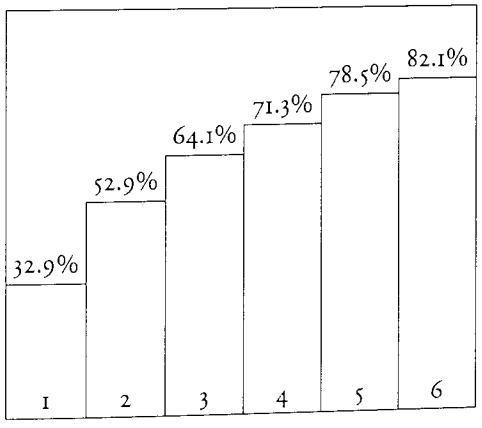


Figure 6.3 – Explaining power of one to six factors of the mail database

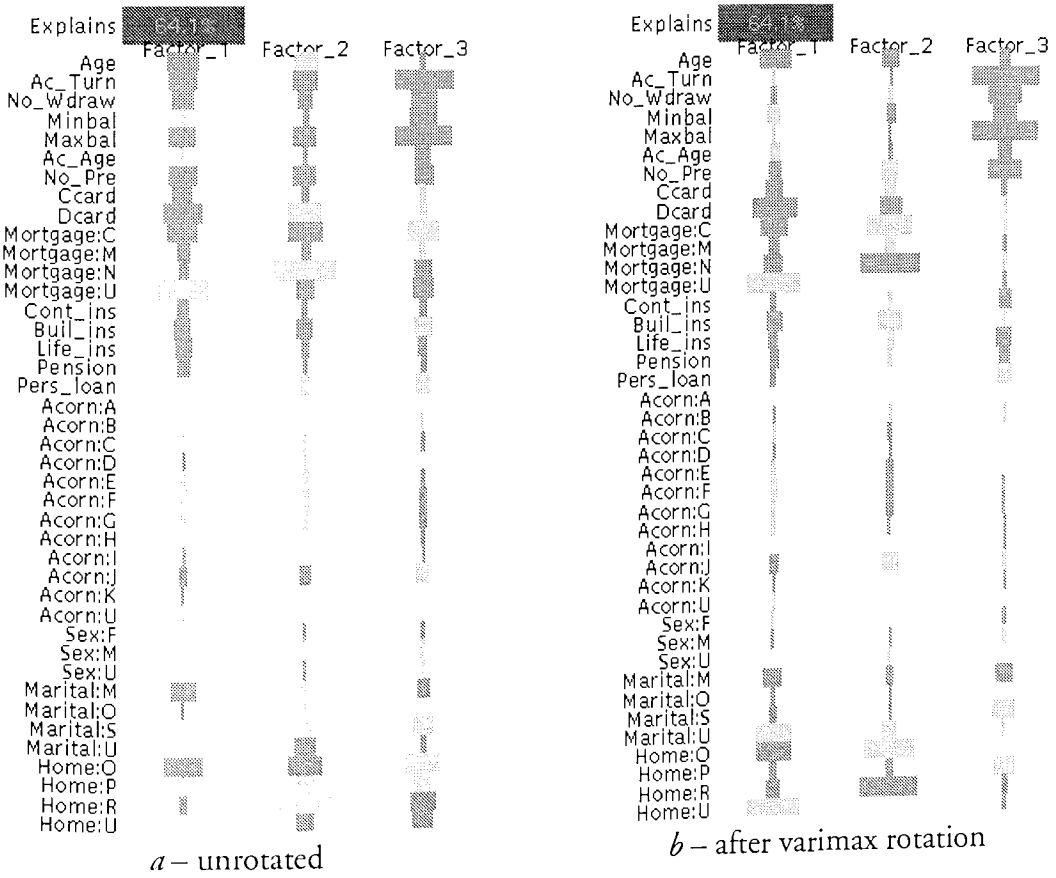
To determine how many factors to use, the 'explaining power' of one to six factors was measured (by using MADEN to find each number in turn), as shown in figure 6.3. Three factors were chosen, this being the approximate 'elbow' in the graph.

Figure 6.4*a* shows the three factors which explain 64.1% of the covariance in the mail database. As expected, there is an extreme similarity to the first three principal component axes shown in figure 6.1, because the data is standardised before applying PCA.

Of greater interest is figure 6.4*b* which shows the three factors after varimax rotation (and flipping of the second factor). Several fields, for example `Ac_Turn`, `Maxbal` and `Home:U`, show how the rotation operation has identified factors which share components in a particular field and 'concentrated' the component into one factor.

As with the PC axes, interpretation of the rotated factors is possible:

- Factor one groups older customers with debit cards with known marital and home status, particularly home owners.
- Factor two clearly separates out the customers who rent their homes.
- Factor three is definitely a 'wealth index', more 'concentrated' than the PC version.



a – unrotated *b* – after varimax rotation

Figure 6.4 – Three factors of the mail database

Figure 6.5 shows an overview of the estimated factor loadings using the three rotated factors. Nine distinct groups can clearly be seen in the PF_1–PF_2 plot. The PF_3 'wealth index' provides little discrimination between groups, except for a small group of very low wealth.

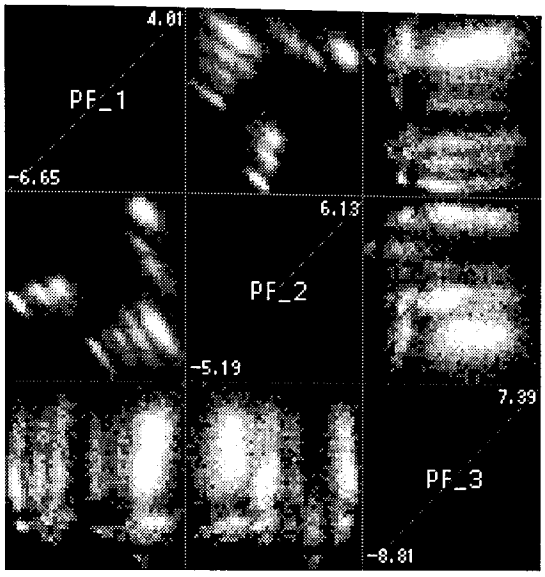


Figure 6.5 – Overview of the (estimated) factor scores of the mail database, using three rotated factors

6.8.1.3 Directed principal component analysis

Applying DPCA to the mail database generates a set of axes, the first four of which are shown overleaf in figure 6.6. The eigenvalues show that one axis is a lot more significant than the remainder. This axis is therefore highly correlated with Response, and is dominated by the Life_ins field, which in chapter 3 was seen to be strongly linked to Response.

The second DPCA axis picks out young married or single people with credit and debit cards and unknown home and mortgage status; the third axis seems to select customers whose sex is known; the fourth axis separates non-singles who make a large number of withdrawals.

An overview showing the database projected onto these four axes, with Response overlaid, is shown in plate 6.2 on the following page.

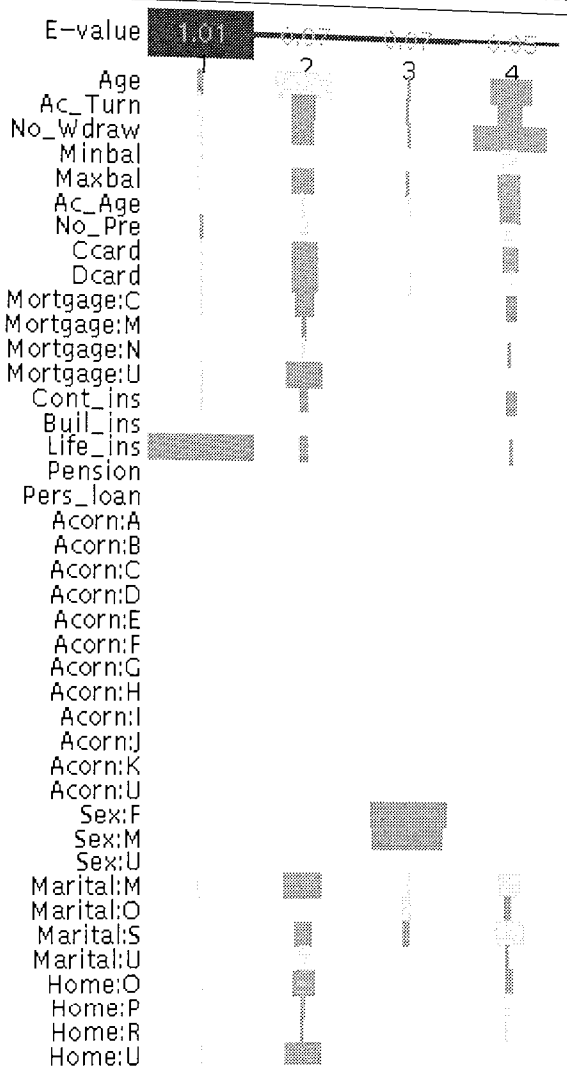


Figure 6.6 – First four DPCA axes of the mail database

Examining plate 6.2 shows a very different view of the data from that seen in figure 6.2 on page 194. As shown on the previous page, three binary fields (`Life_ins`, `Sex:M` and `Sex:F`) contribute strongly to PC_1 and PC_3, since these fields have a relatively high correlation with `Response`. Their contribution to the projection is therefore increased, resulting in widely spaced groups of data, particularly visible in the plot of PC_1 against PC_3.

The effect of DPCA on the distribution of responders is not as dramatic as might be hoped. Clearly PC_1 separates out a large number of non-responders, due to `Life_ins`, but the other three fields show only a little discrimination between red and blue areas.

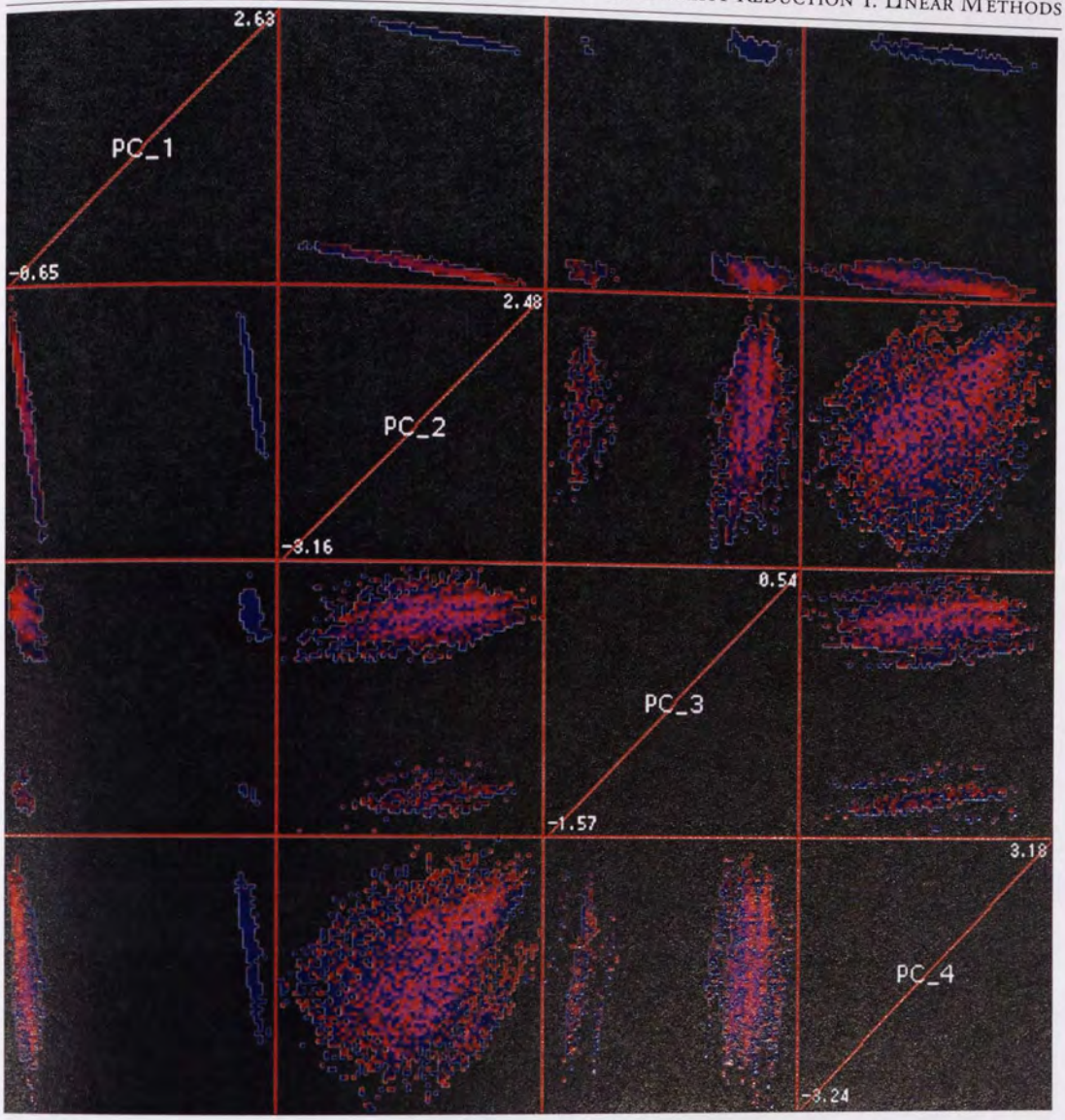
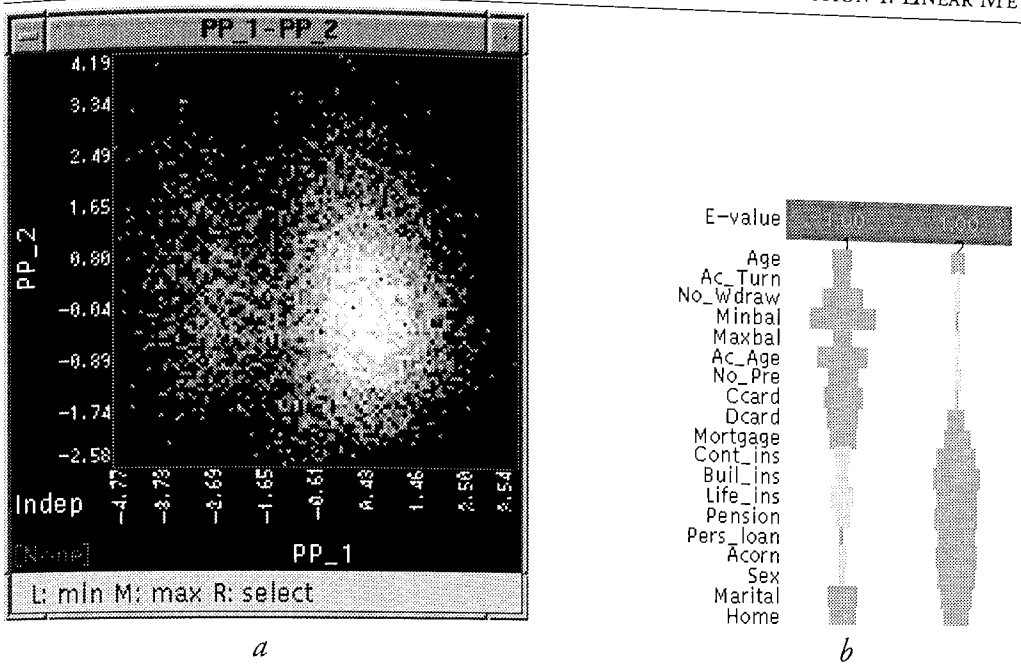


Plate 6.2 – Mail database projected onto four DPCA axes, with Response overlaid

6.8.1.4 Projection pursuit

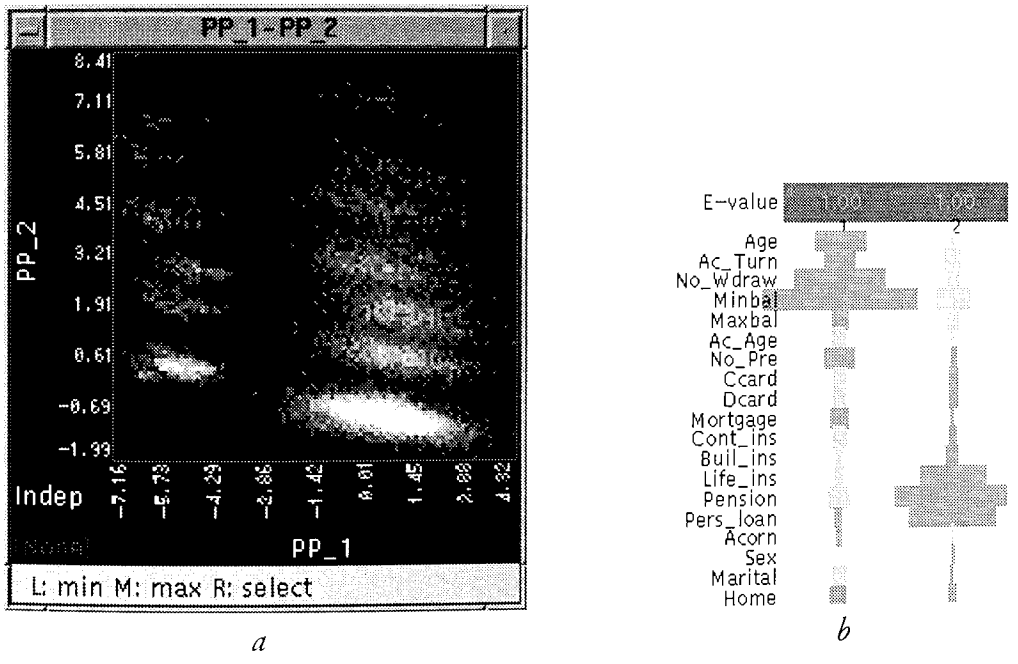
Projection pursuit was attempted on the mail database using the three 'quick' PP indices. The 'holes' index would not converge, but the other two gave acceptable results.

PP projections with optimal central mass and skew indices are shows in figures 6.7 and 6.8 respectively (overleaf), along with the axes used to general the plots. Plate 6.3 on the following page shows the projections with Response overlaid.



a *b*

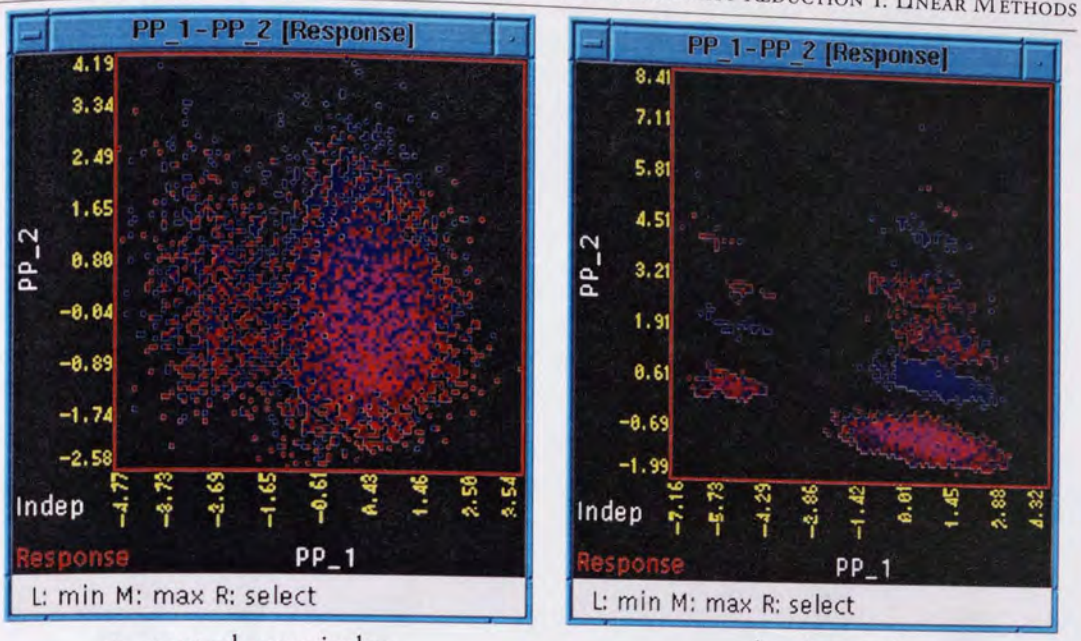
Figure 6.7 – Result of PP on the mail database using central mass index



a *b*

Figure 6.8 – Result of PP on the mail database using skew index

As expected, the central mass index results in a fairly dull projection, with most of the data clumped into an elliptical cluster to the right of the plot, though a number of records are projected outside this cluster. Some patterns in the distribution of responders can be seen, notably an area of high response towards the lower right of the plot, but generally the plot is uninteresting.



a – central mass index

b – skew index

Plate 6.3 – Results of PP on the mail database, with Response overlaid

The skew index, however, finds a very interesting projection of the data, containing at least twelve distinct clusters of data. Overlaying Response shows that some clusters contain no responders at all (once again, due to the Life_ins field), some are fairly homogenous (particularly the dense cluster at the lower right of the plot), and some have a more interesting spread of responders. It appears that the skew index has succeeded in automatically finding an 'interesting' view of the data.

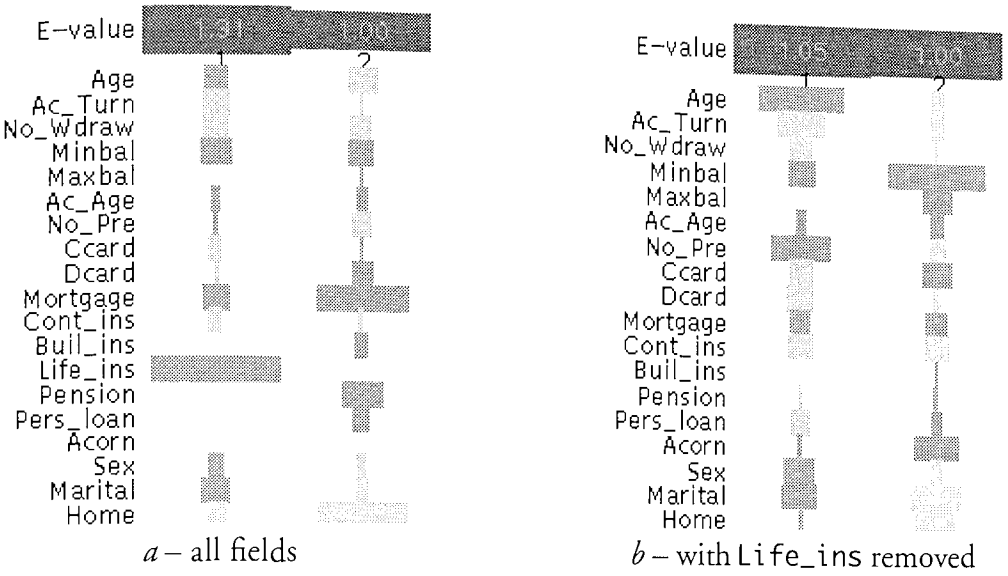
6.8.1.5 Linear discriminant analysis

Figure 6.9*a* overleaf shows the result of applying LDA to the mail database. Note that the second axis has unit $F-$ because the mail database has only two classes between which to discriminate, there is only one true linear discriminant axis, and the other is simply the variance-maximising axis orthogonal to the genuine discriminant axis.

The LDA confirms once again that the Life_ins field is by far the most important field for separating the reponders from the non-responders. In addition, it seems that customers with low age, high turnover, high number of withdrawals and low minimum balance are more likely to respond. The discrimination factor F is 1.31, indicating that the class means have 31% more variance than the average variance within each class.

To see how much separation could be achieved without the aid of Life_ins, the Life_ins field was clipped out of the database and another LDA was performed, generating the discriminant axis shown in figure 6.9*b*, which has an F of only 1.05.

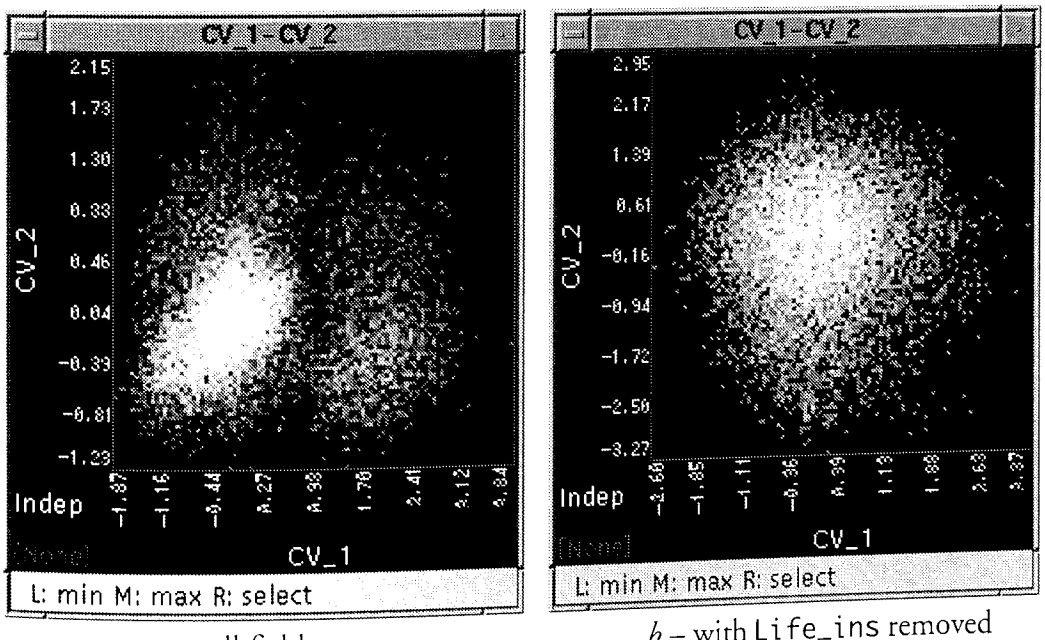
Evidently there is little intrinsic linear separation between reponders and non-responders.



a – all fields *b* – with `Life_ins` removed

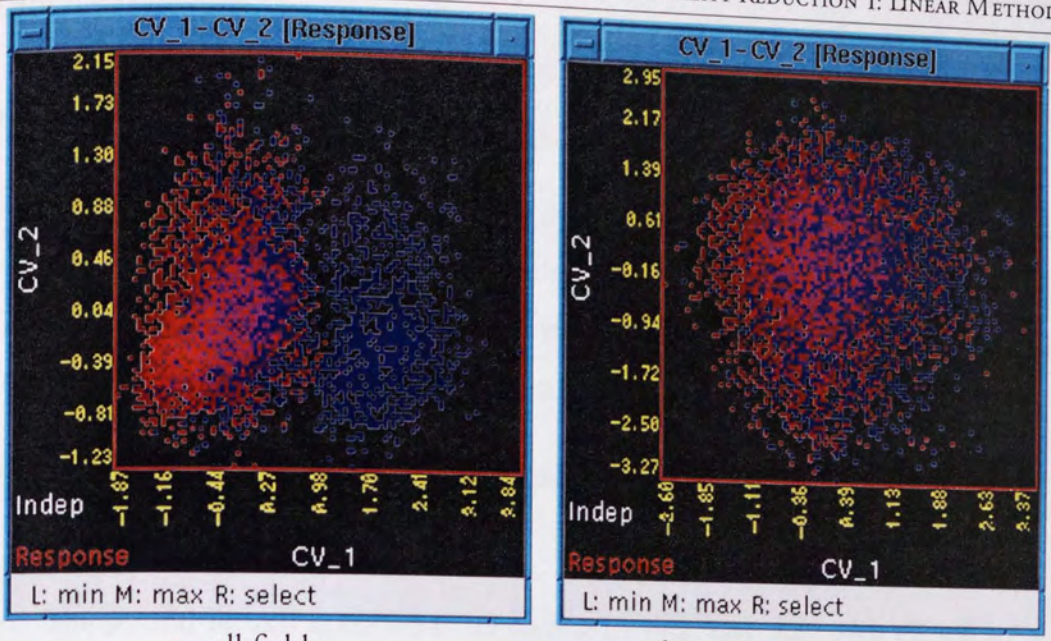
Figure 6.9 – Linear discriminant axis of the mail database

Figure 6.10 shows the projections onto the linear discriminant axis of the database, both with and without the `Life_ins` field. The separation of reponders from non-responders is illustrated in plate 6.4overleaf, which shows both these canonical variate plots with the `Response` field overlaid. Without the information from the `Life_ins` field, the separation is dramatically less noticeable.



a – all fields *b* – with `Life_ins` removed

Figure 6.10 – Canonical variate of the mail database



a – all fields

b – with `Life_ins` removed

Plate 6.4 – Canonical variate of the mail database, with `Response` overlaid

6.8.2 Finance database

6.8.2.1 Principal component analysis

Figure 6.11 shows the first five PC axes of the finance database. The first two axes are roughly of equal importance, significantly more so than the remaining axes. Full interpretation of the axes is prevented by the lack of knowledge regarding field meanings, but a description of the first two axes can be given:

- The first PC axis separates customers who have unknown t3, q1, q2 and q3, answered no to q4 and yes to q5.
- The second PC axis again separates customers who have unknown t3, q1, q2 and q3, but who answered yes to q4 and no to q5.

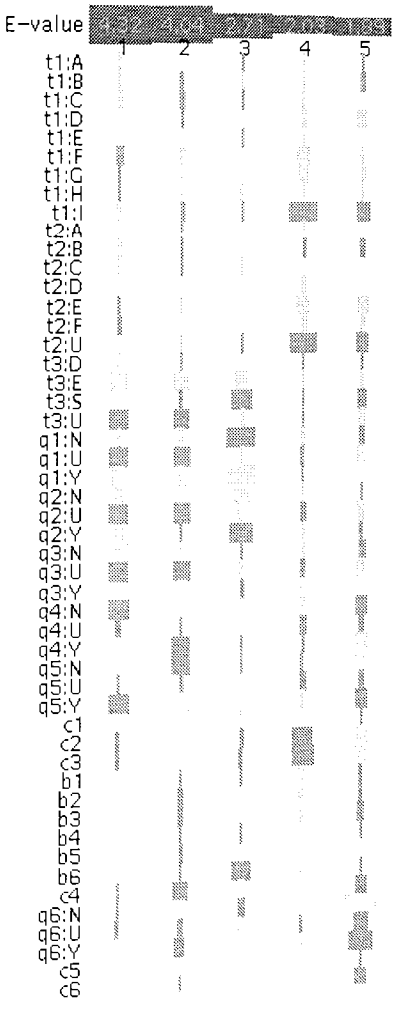


Figure 6.11 – First five principal components of the finance database

An overview of the projection onto the first five PC axes is shown in figure 6.12. Clustering within the data is evident in most of the density plots, particularly PC_1-PC_2 and PC_2-PC_3.

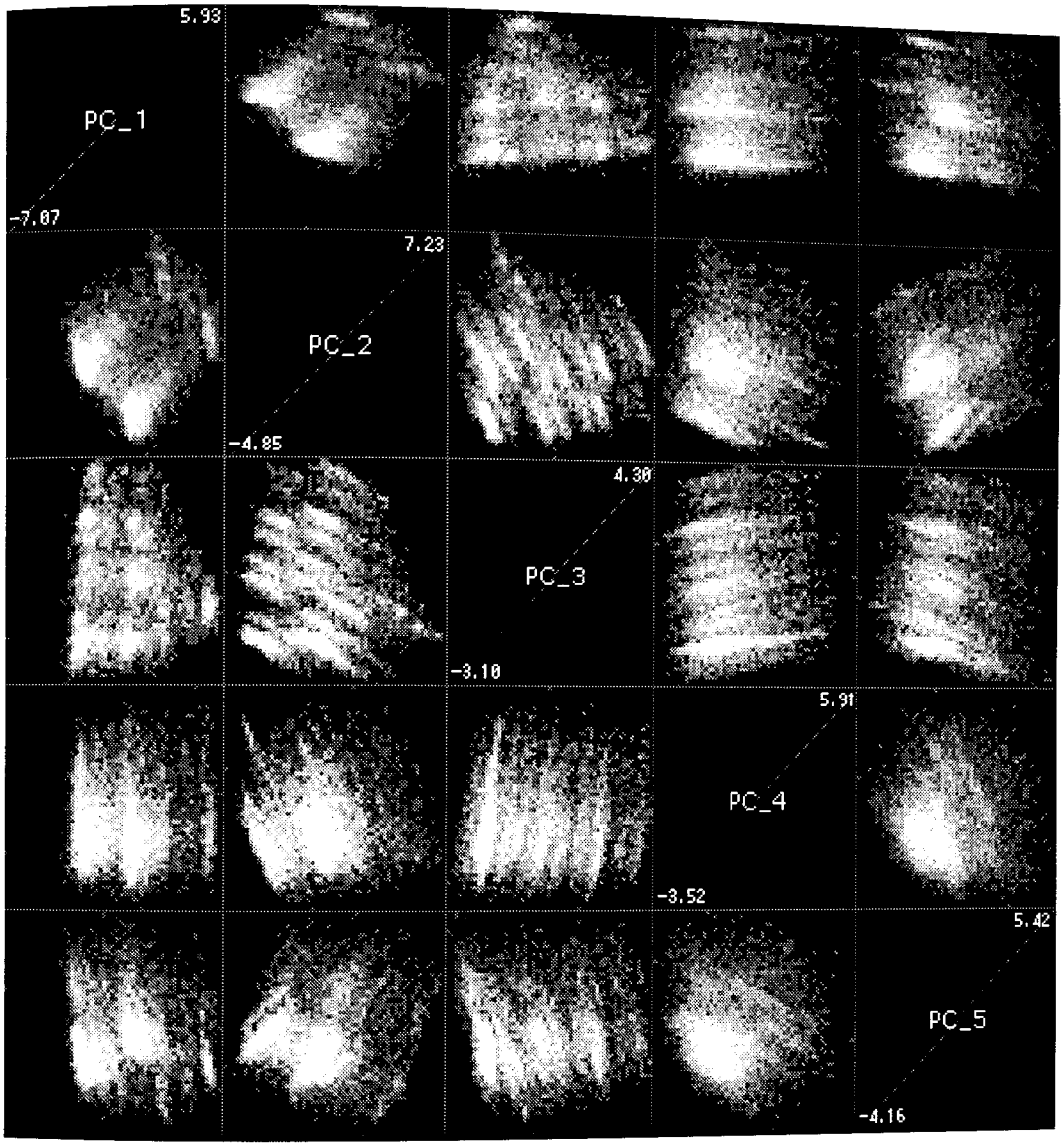


Figure 6.12 – Overview of finance database projected onto five principal components

6.8.2.2 Directed principal component analysis

The first four DPCA axes of the finance database are shown in figure 6.13. Unlike the analysis of the mail database, the eigenvalues do not suddenly drop to very small amounts after the first axis, but yet again, interpretation of the axes is impossible.

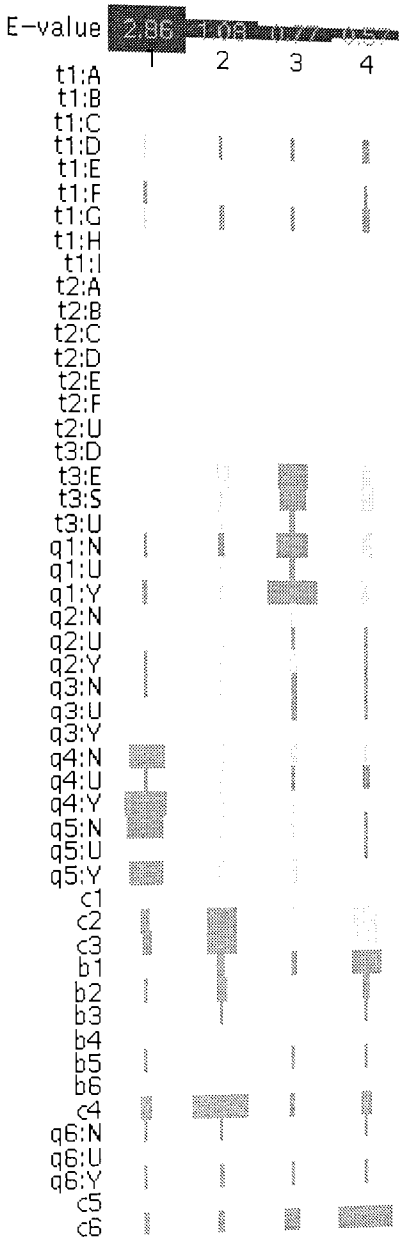


Figure 6.13 – First four DPCA axes of the finance database

Plate 6.5 overleaf shows the projection of the database onto these DPCA axes, with the `response` field overlaid. It is immediately clear from the colouring that the analysis has resulted in a set of axes which are all highly correlated with the response field.

In addition, clustering is evident on the `PC_3` projections, and a number of outlying points can be seen along `PC_1` which might well prompt further investigation.

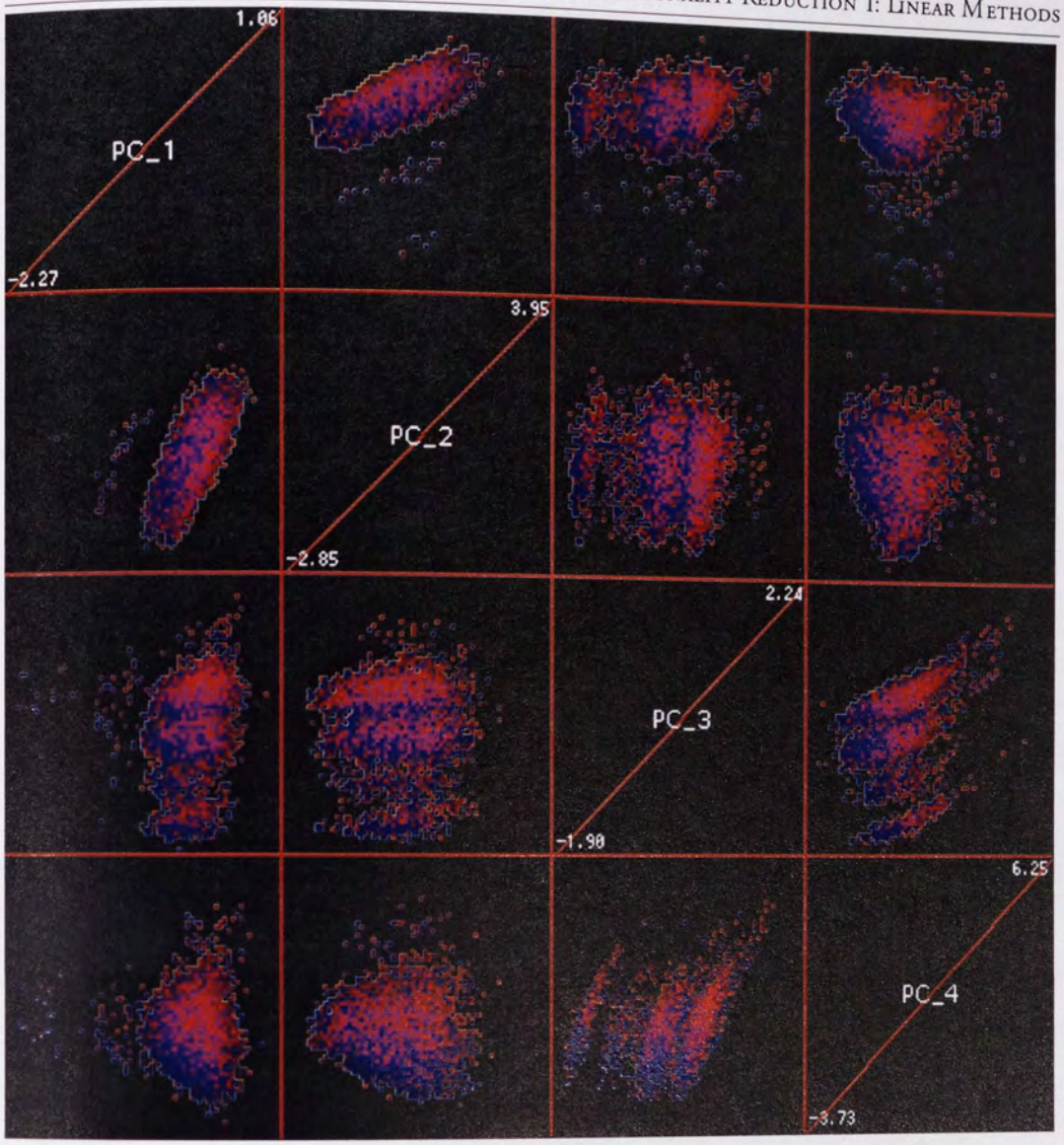
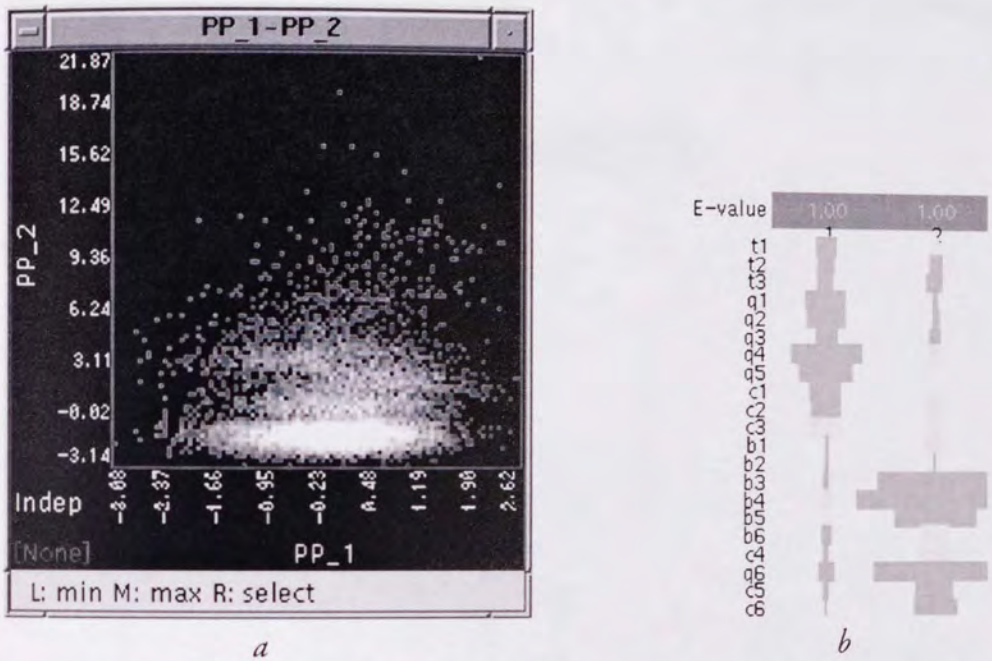


Plate 6.5 – Finance database projected onto four DPCA axes, with response overlaid

6.8.2.3 Projection pursuit

Only the skew PP index gave a result with the finance database; the holes and central mass indices failed to converge. Figure 6.14 and plate 6.6 show the resulting axes and projections. The plot is less 'interesting' that the one generated from the mail database, but shows some clear outliers, and a set of at least three clusters, which appear elongated only because the outlying points dramatically change the scale of the PP_2 axis.



a *b*

Figure 6.14 – Result of PP on the finance database using skew index

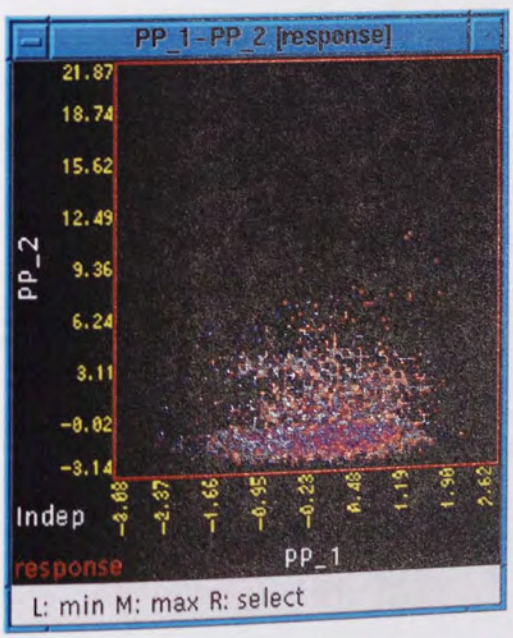


Plate 6.6 – Result of PP on the finance database using skew index, with response overlaid

6.8.2.4 Linear discriminant analysis

An LDA was performed on the finance database, in an attempt to separate the reponders from the non-responders. The resulting axis is shown in figure 6.15. The F value of 1.11 shows that some separation is possible, but not a lot. Plate 6.7 shows the canonical variate generated from the linear discriminant axis, plotted against the 'dummy' canonical variate, with the response overlaid. The blue non-responders can be seen clustered at the left side, but the plot demonstrates that it is very hard to separate the two groups using linear methods. Interestingly, the LDA 'optimal' separation appears to be less than that achieved by the fourth DPCA axis, seen in plate 6.5.

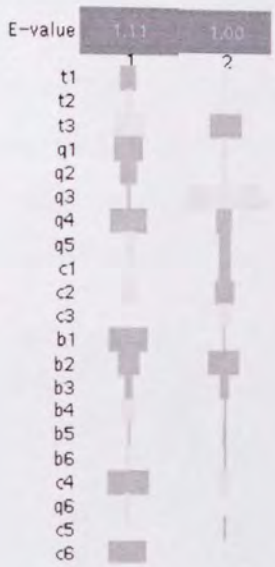


Figure 6.15 – Linear discriminant axis of the finance database

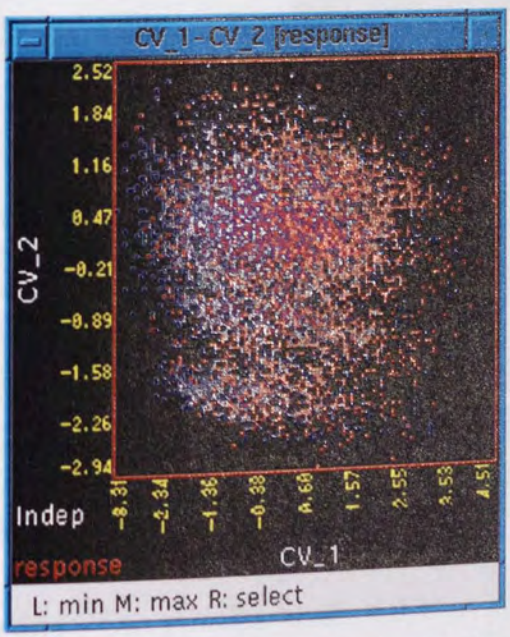
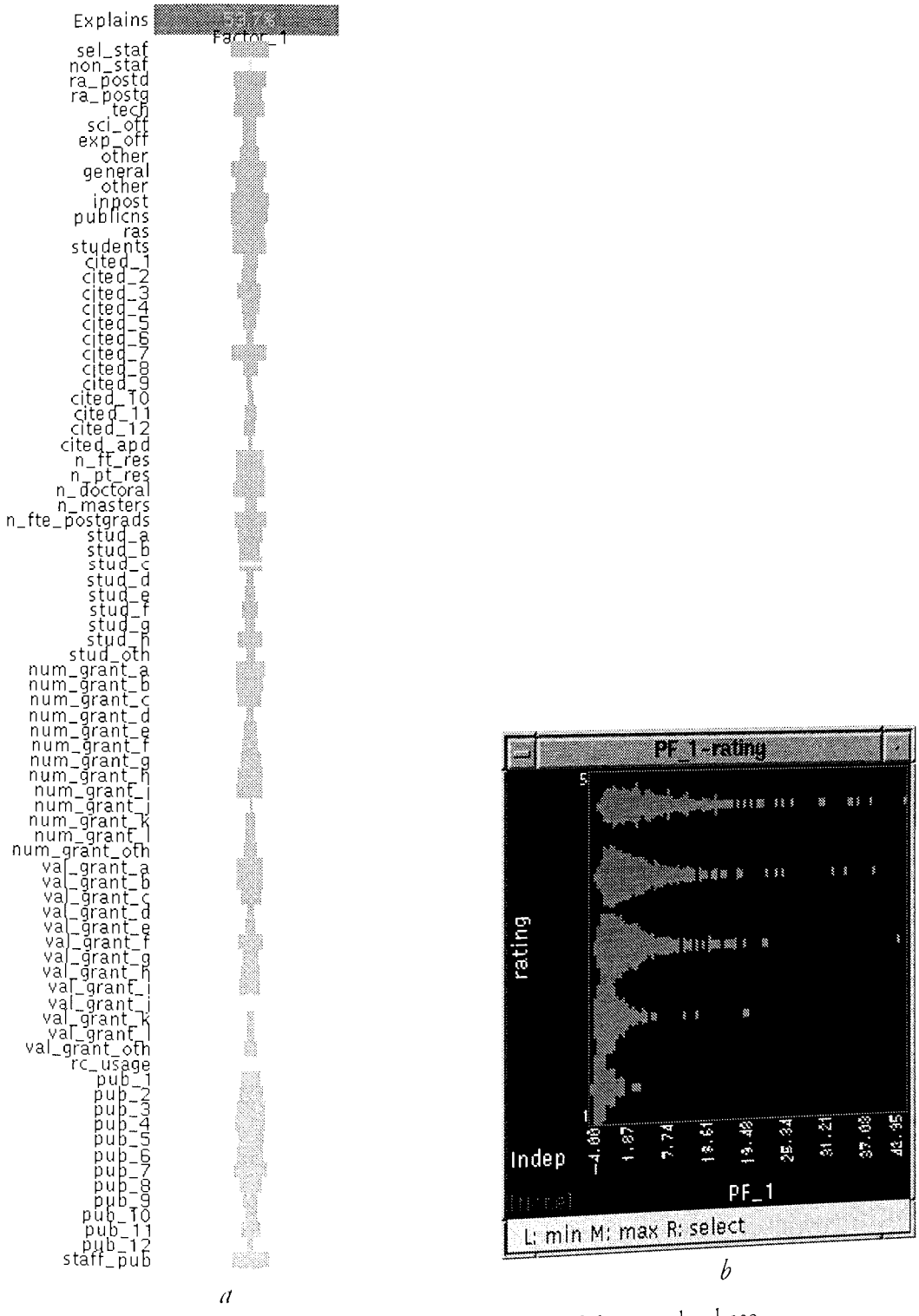


Plate 6.7 – Canonical variate of the finance database with response overlaid

6.8.3 RAE database

6.8.3.1 Factor Analysis

Experimenting with factor analysis of the RAE database revealed that one factor explained 53.7% of the covariance. This factor is shown in figure 6.16*a*.



a *b*

Figure 6.16 – One principal factor of the RAE database, and the plot of its factor scores against `rating`

The factor has fairly large positive components in virtually all fields, and can therefore be described as a measure of the 'size' of a department. Departments which have a lot of staff, or produce a lot of publications (or receive many grants, etc) score highly on this factor. Figure 6.16*b* shows the plot of the factor scores against `rating`, demonstrating a clear correlation: 'larger' departments are more likely to receive a high research rating than 'small' ones, as measured on the 'size' factor score.

Two fields have very small negative loadings on the factor: `non_staf` and `val_grant_j`. The former is the number of staff not selected to contribute to the research assessment, which is understandable, the latter the value of grants awarded from PCFC/NAB initiatives, which is less obvious. Maybe these grants are generally made to small departments.

Having determined a measure of the 'size' of a department, and seen that this measure is strongly correlated with the rating awarded, it seemed reasonable to standardise the database by this measure, by using the divide operation which was introduced in chapter 4 to divide the data by the first factor scores.

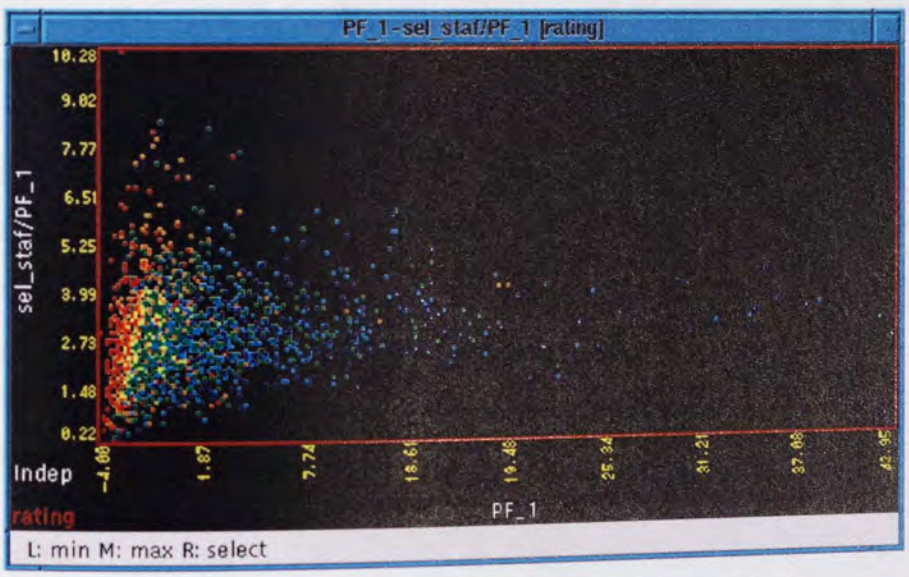


Plate 6.8 – Factor scores of the first factor of the RAE database plotted against standardised `sel_staf`, with `rating` overlaid.

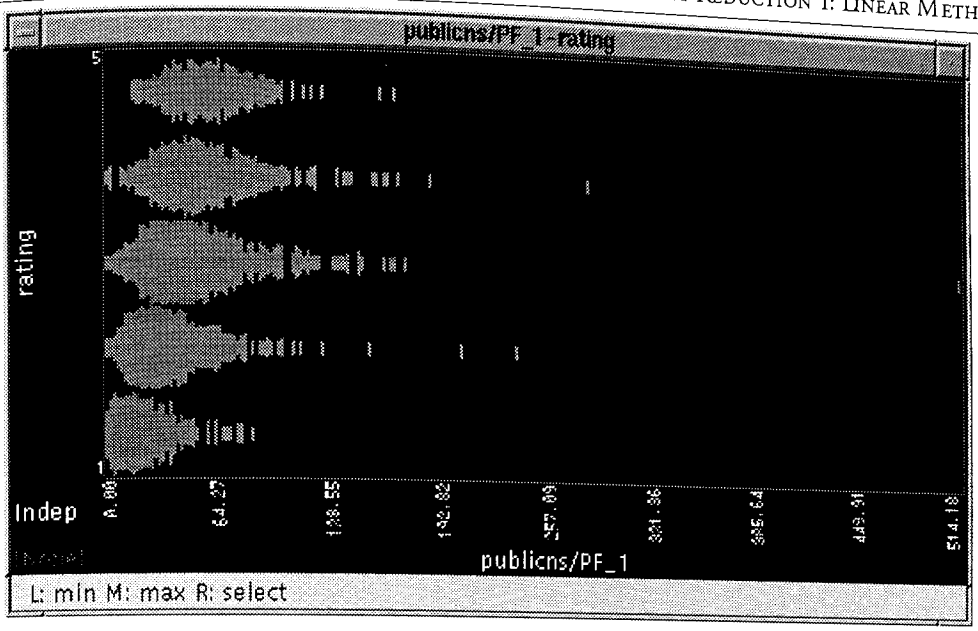
Plate 6.8 shows the factor scores plotted against the newly-standardised field `sel_staf/PF_1`, which measures the relative number of selected staff for the 'size' of the department. The rating is overlaid, and a clear pattern of rating locations can be seen. As previously noted, high `PF_1` generally results in a high rating. However, it seems that `sel_staf/PF_1` is slightly negatively correlated with rating: if a department has more than the average number of selected staff for its 'size', it has a fair chance of being awarded a lower research rating than similarly-'sized' departments.

Figure 6.17 overleaf shows three enlargements from the standardised database, each plotted against `rating`:

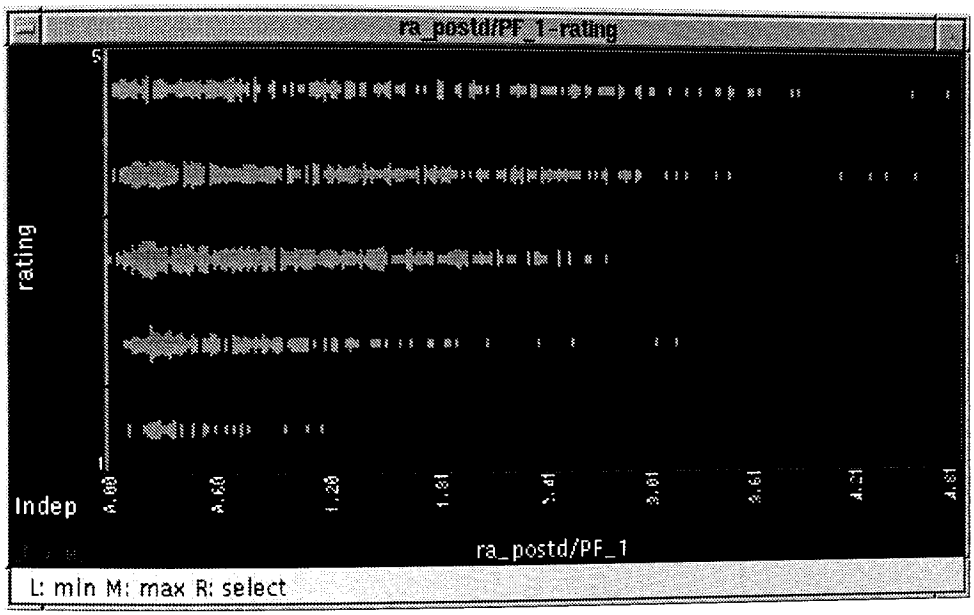
- Figure 6.17*a* shows how the relative number of publications produced by a department affects its research rating. Once again, there is a strong relationship: produce more publications than similarly-'sized' departments and a higher rating is more likely. Using highlighting, the three outlying departments on the right side were identified as: rating three, Music at the University of Manchester; rating four, Music at the University of Leeds; rating two: Art & Design at the University of East London. These departments evidently produced far more publications than their 'size' would suggest.

- `ra_postd`, the number of postdoctoral researchers, is also correlated with `rating`, as figure 6.17*b* reveals. The rating three outlier, which failed to get the high rating suggested by its very large relative number of postdocs, was Biochemistry at University College London.

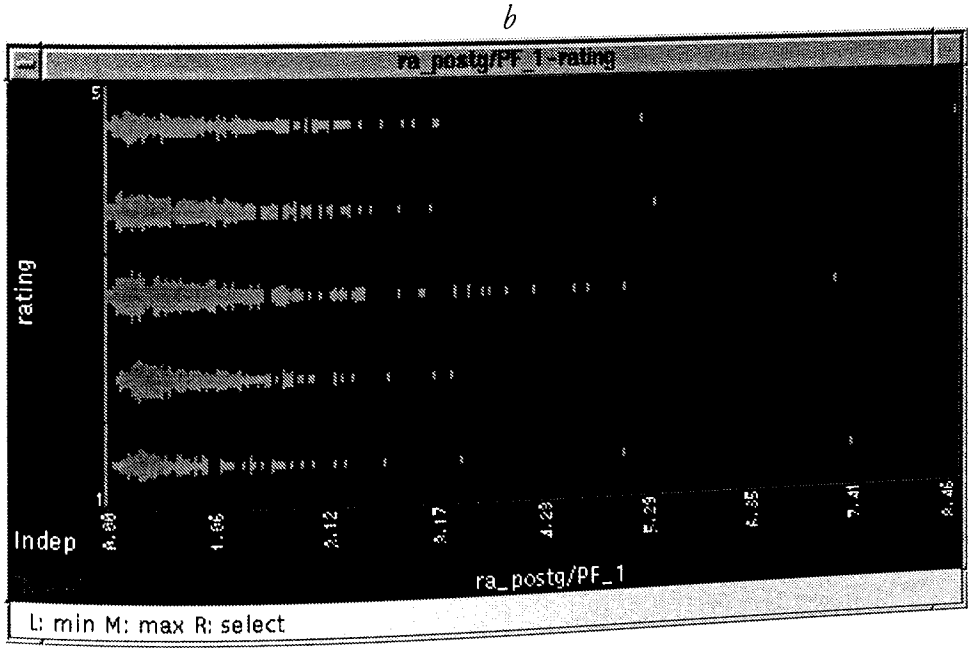
- However, it seems that postgraduate researchers do not make a great difference to the research rating. Figure 6.17*c* shows, if anything, a slight negative correlation. Maybe postgrads prevent real research taking place! The three outliers with abnormally high number of postgraduate researchers for their size are: rating one, Biological Science at the University of East London; rating three, Pre-clinical Studies at Imperial College; rating five, Chemistry at the University of Cambridge.



a



b



c

Figure 6.17 – Relationships between standardised variables and `rating`

6.8.3.2 Directed principal component analysis

Figure 6.18 below shows the first three DPCA axes of the RAE database.

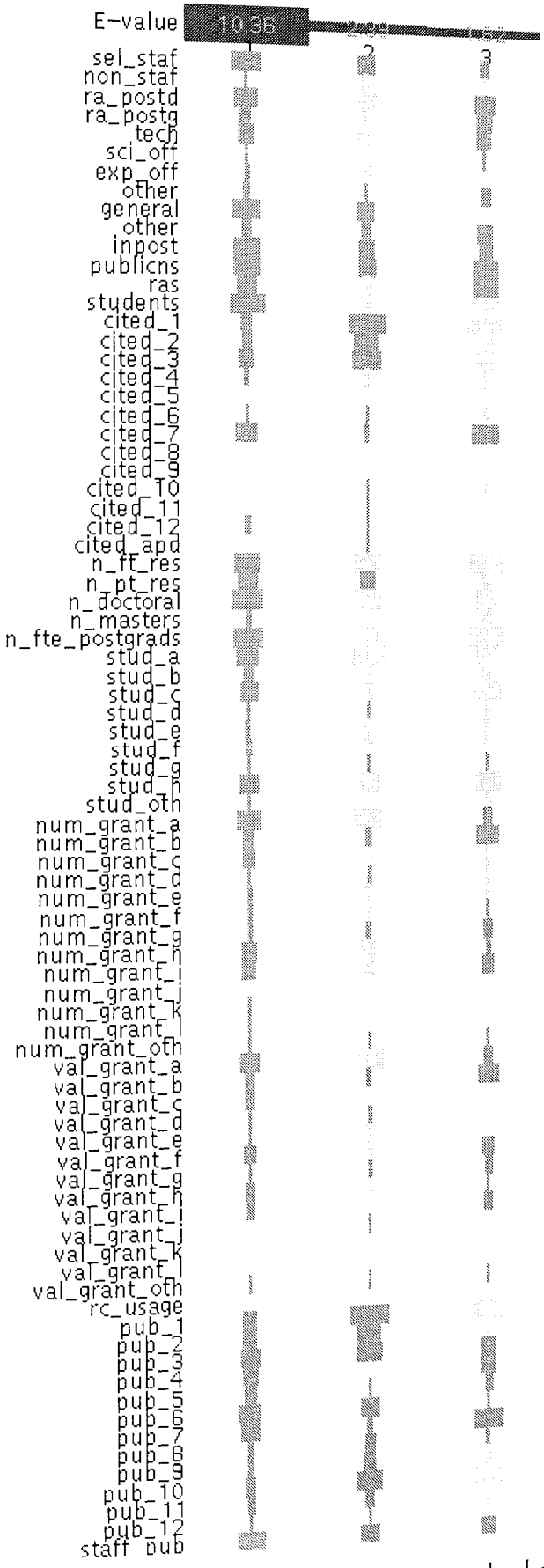


Figure 6.18 – First three DPCA axes of the RAE database

No clear features emerge from examining the DPCA axes, though the variation in the pub_n components is interesting: PC_1 has all the publication types weighted approximately equally, PC_2 has a different distribution in which pub_4 (refereed conferences) has a small negative weighting, and PC_3 has strong negative weightings for pub_1 (authored books) and pub_10 (reviews of academic books).

Plate 6.9 shows the projection onto the DPCA axes, with Rating overlaid and also shown in the overview, to investigate correlations with the rating.

All the density plots show that the highly-rated departments tend to be separated from the lower-rated ones, which are densely clustered. PC_1 can be seen to be well correlated with Rating, though there is little separation between the ratings for the vast majority of departments, but the other two axes show little discriminatory power.

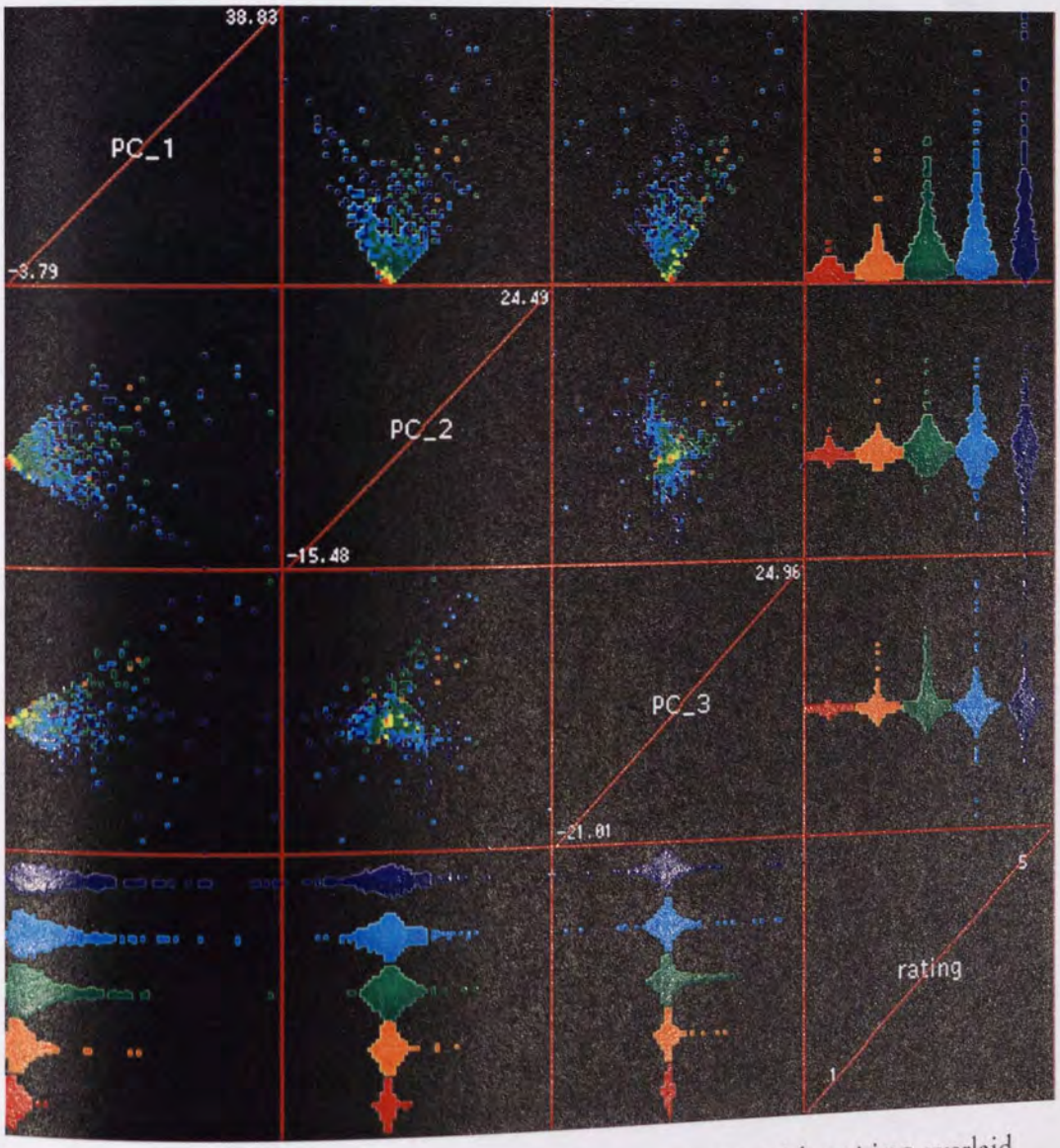
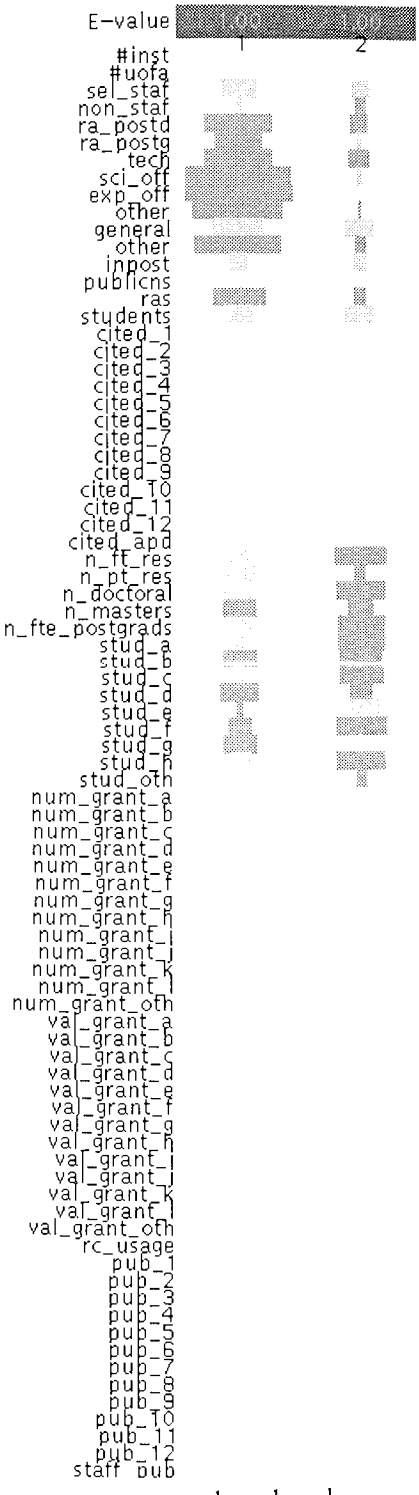


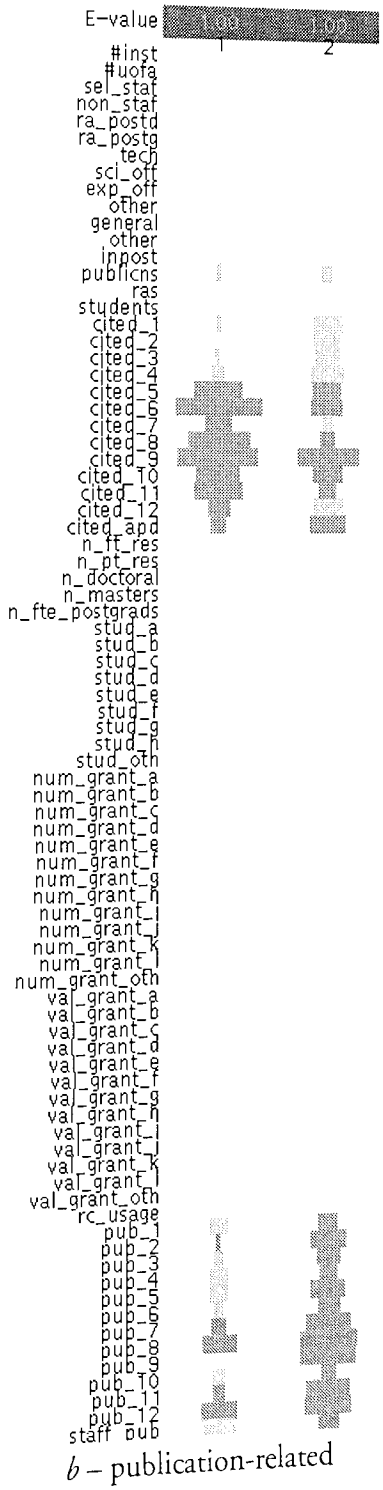
Plate 6.9 – RAE database projected onto three DPCA axes, with rating overlaid

6.8.3.3 Projection pursuit

Projection pursuit was prohibitively slow on the RAE database, due to the large number of fields. To overcome this, separate PP optimisations were carried out using only the 'people-related' or 'publication-related' fields – i.e. fields related to numbers of people or numbers of publications.



a – people-related

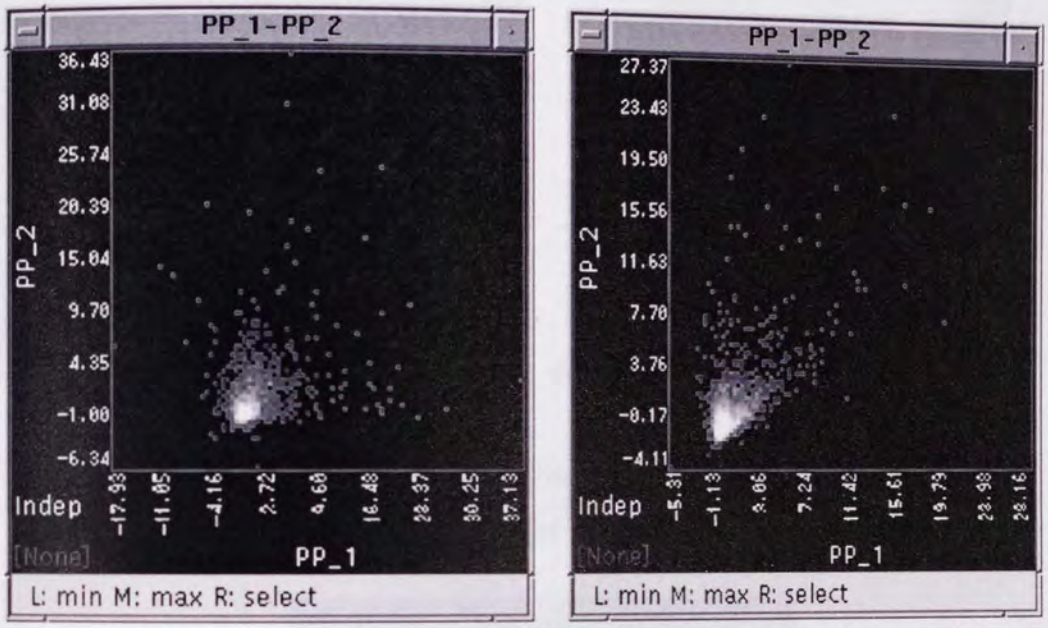


b – publication-related

Figure 6.19 – PP axes of the RAE database using skew index

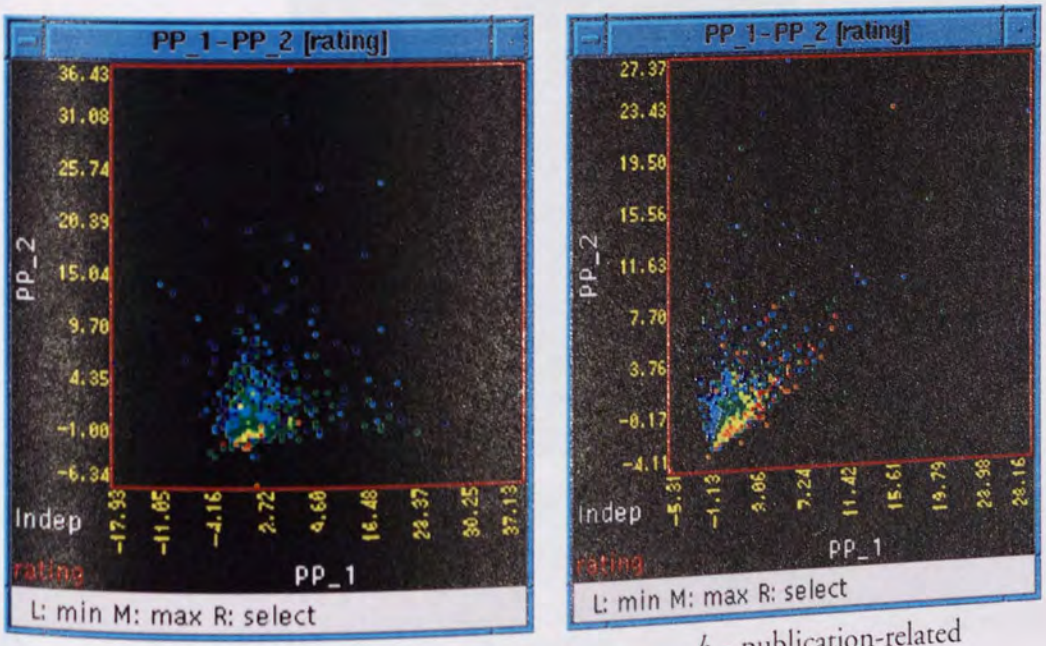
Figure 6.19 shows the optimal skew index PP axes of the database, modifying only the people- or publication-related fields (the fields can be identified from the figure). The resulting projections are shown in figure 6.20, and in plate 6.10 with `rating` overlaid.

Neither projection is particularly interesting; both resemble typical plots of the database onto two of its original fields. Evidently there are outliers in both projections which could be identified and studied, but since the projections do not appear to reveal any interesting features of the database, this was not carried out.



a – people-related *b* – publication-related

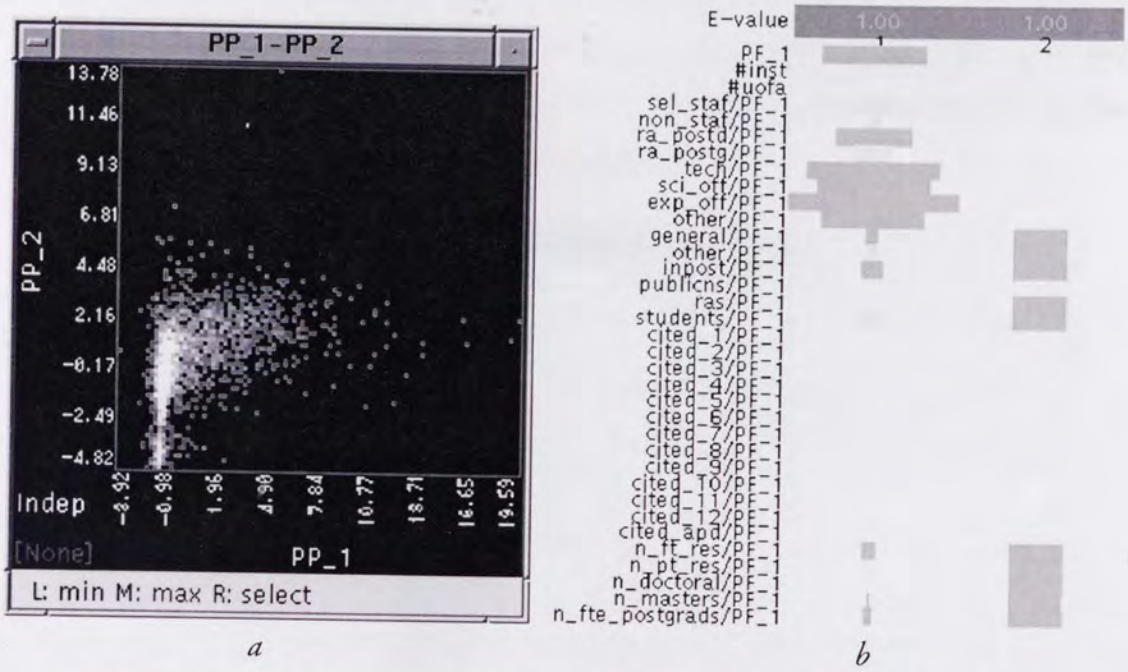
Figure 6.20 – PP projections of the RAE database using skew index



a – people-related *b* – publication-related

Plate 6.10 – PP projections of the RAE database using skew index, with `rating` overlaid

Before dismissing PP as a useful technique for the RAE database, it was applied to the people-related fields of the standardised database created by dividing by the first principal factor. The results are shown below in figure 6.21 and plate 6.11.



a *b*

Figure 6.21 – Results of PP on the people-related standardised RAE database using skew index

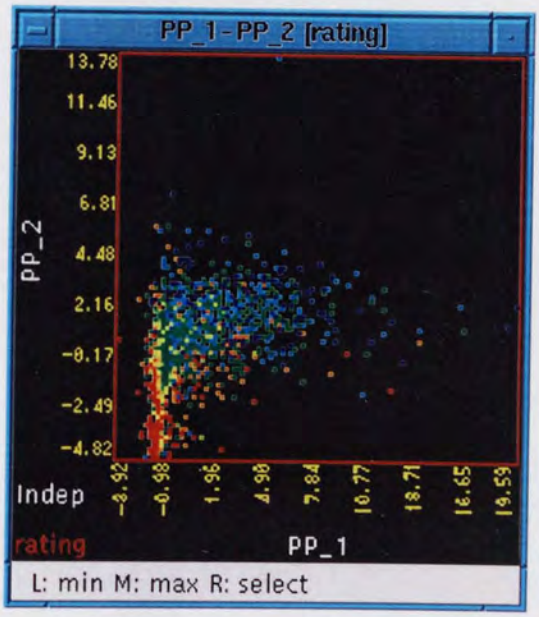


Plate 6.11 – Result of PP on the people-related standardised RAE database using skew index, with `rating` overlaid

The projection is definitely more 'interesting' than the previous ones, with each rating group confined to a fairly well defined area, particularly the low-rated departments which are almost in a separate cluster at the bottom left.

6.8.3.4 Linear discriminant analysis

Linear discriminant analysis of the RAE database generated four axes, since there are five classes in the database. The first axis had a significantly higher separation ratio (0.88) than the other axes. The first two axes are shown in figure 6.22. The axes are strikingly similar, sharing large components in the `n_ft_res`, `n_pt_res` and `n_fte_postgrads` fields and differing only in relatively small components in the other fields. They are also clearly not orthogonal.

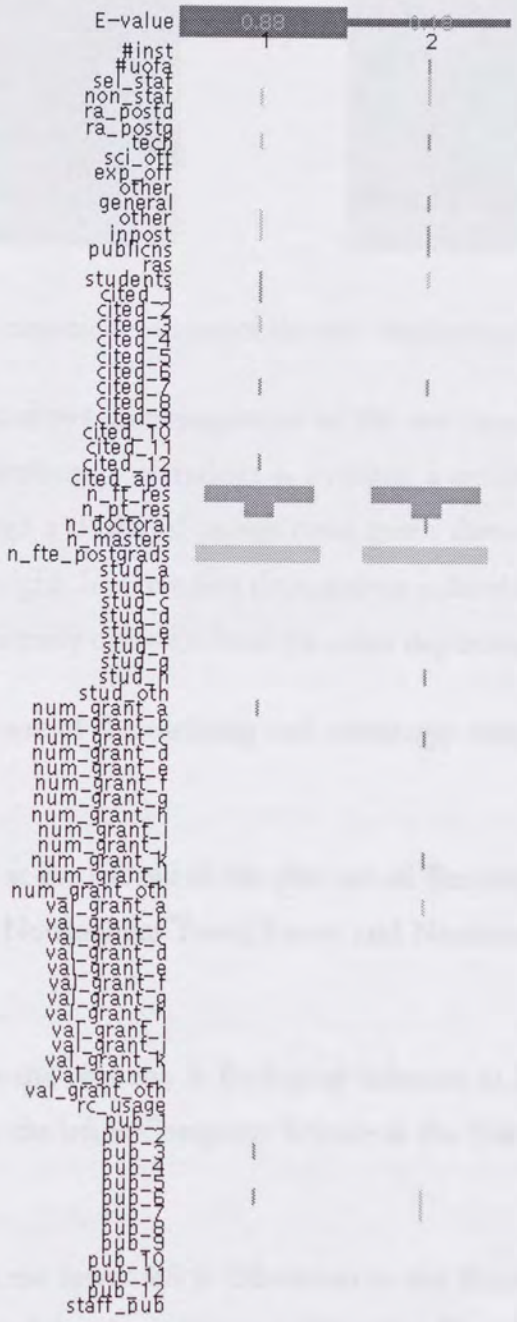
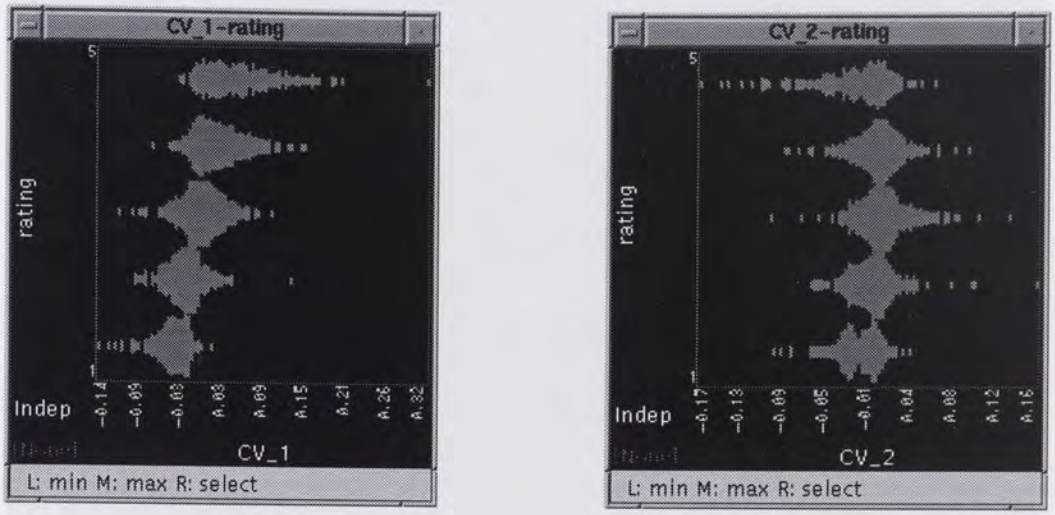


Figure 6.22 – Two most significant linear discriminant axes of the RAE database

Figure 6.23 illustrates the class separation achieved by the LDA. The plots show the first two canonical variates plotted against `rating`. `CV_1` is roughly proportional to `rating` whereas `CV_2` seems to distinguish the middle ratings from the extremes.



a *b*

Figure 6.23 – Two canonical variates of the RAE database plotted against rating

Plate 6.12 on page 220 shows an enlargement of the two canonical variates, overlaid with `rating`. The distribution of ratings is striking: a smooth curve from the red ones on the left, through a cluster of orange twos, green threes and cyan fours out to blue fives at the lower right. It seems that departments achieving fives (and, to a lesser extent, ones) are significantly different from the other departments.

Using highlighting, some of the outlying and seemingly misplaced points on plate 6.12 were identified:

- The four red ones at the left tail of the plot are all Business & Management, at Thames Valley, Nottingham Trent, Luton and Northumbria at Newcastle Universities.

- The orange two at the very top is Biological Sciences at Brunel University; the outlying two at the left is Computer Science at the University of the West of England.

- The green three at the lower left is Education at the Roehampton Institute; the three at the top of the plot is Hospital Clinical at Dundee.

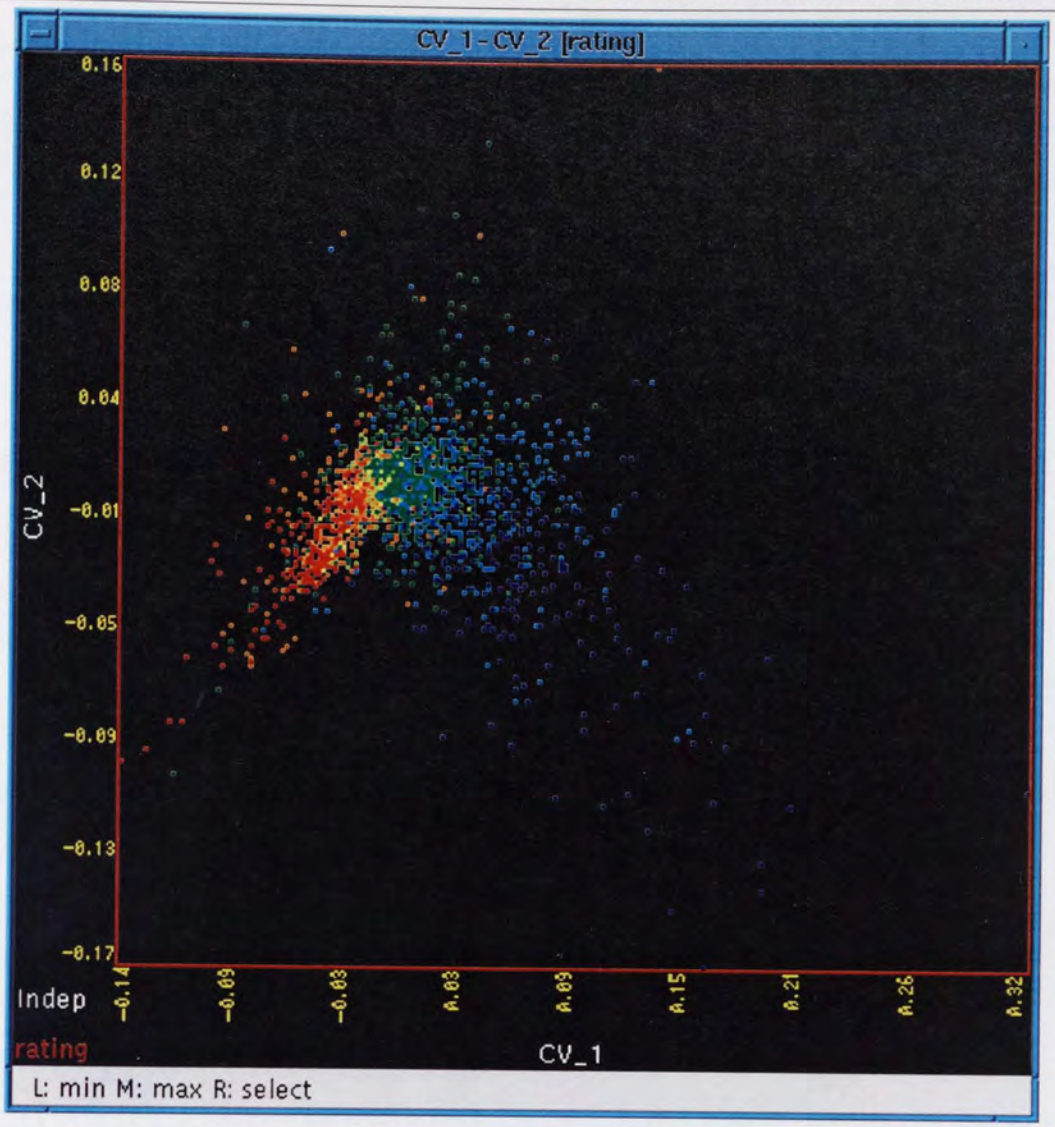
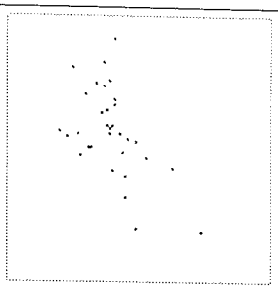


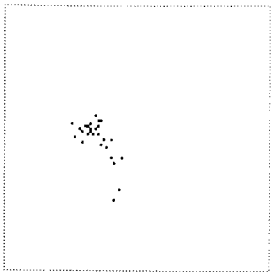
Plate 6.12 – Two canonical variates of the RAE database with `rating` overlaid

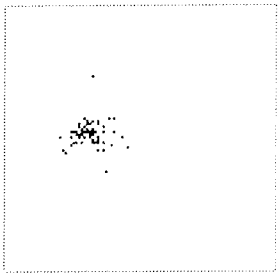
- The cyan four in amongst the red ones is Accountancy at Thames Valley University; the two fours towards the bottom right, among the fives, are Physics at UCL and History at Oxford.

- The blue five on the far right is History at Cambridge; the blue five at the very bottom is Community Clinical at the Institute of Psychiatry.

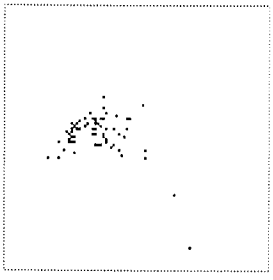
The following two pages use the dependent enlargement feature of MADEN to investigate differences between disciplines (figure 6.24) and institutions (figure 6.25). The figures show small enlargements of the canonical variates plot, with each enlargement dependent on a selection of unit width on the `#uofa` field or `#inst` field.

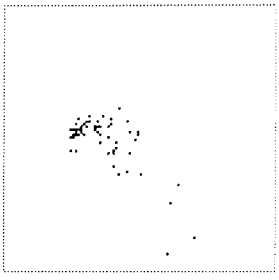

3: Hospital Clinical

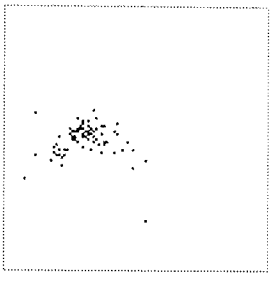

12: Biochemistry

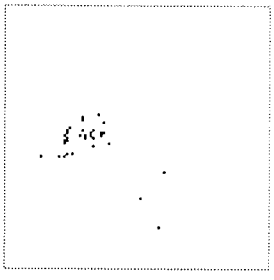

13: Psychology

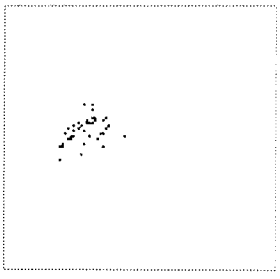

14: Biological Sciences

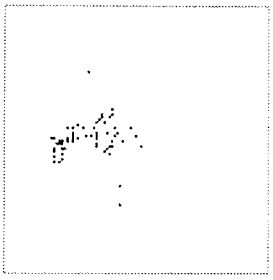

20: Chemistry

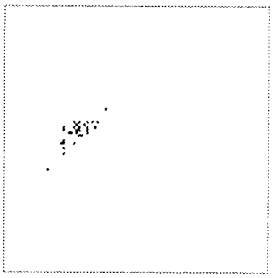

21: Physics

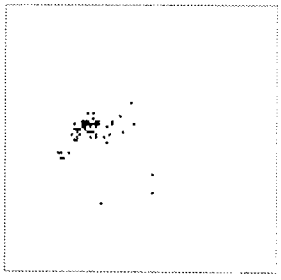

27: Computer Science

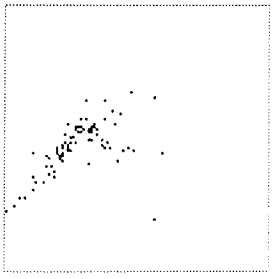

28: General Engineering

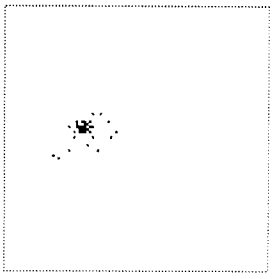

30: Civil Engineering

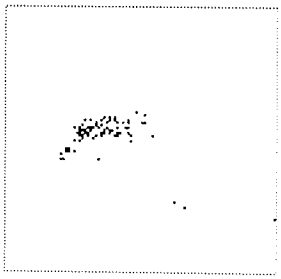

31: Elec Eng

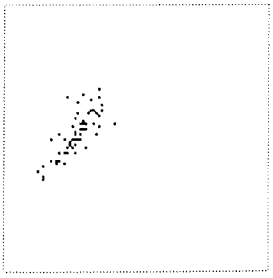

36: Town & Country Planning

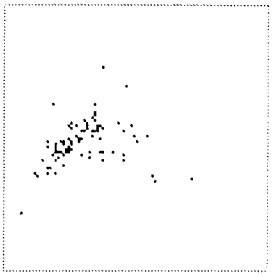

41: Economics


46: Business & Management


54: French


62: History


67: Art & Design


71: Education

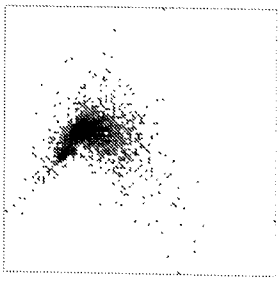

All units

Figure 6.24 – RAE canonical variates, dependent upon unit of assessment

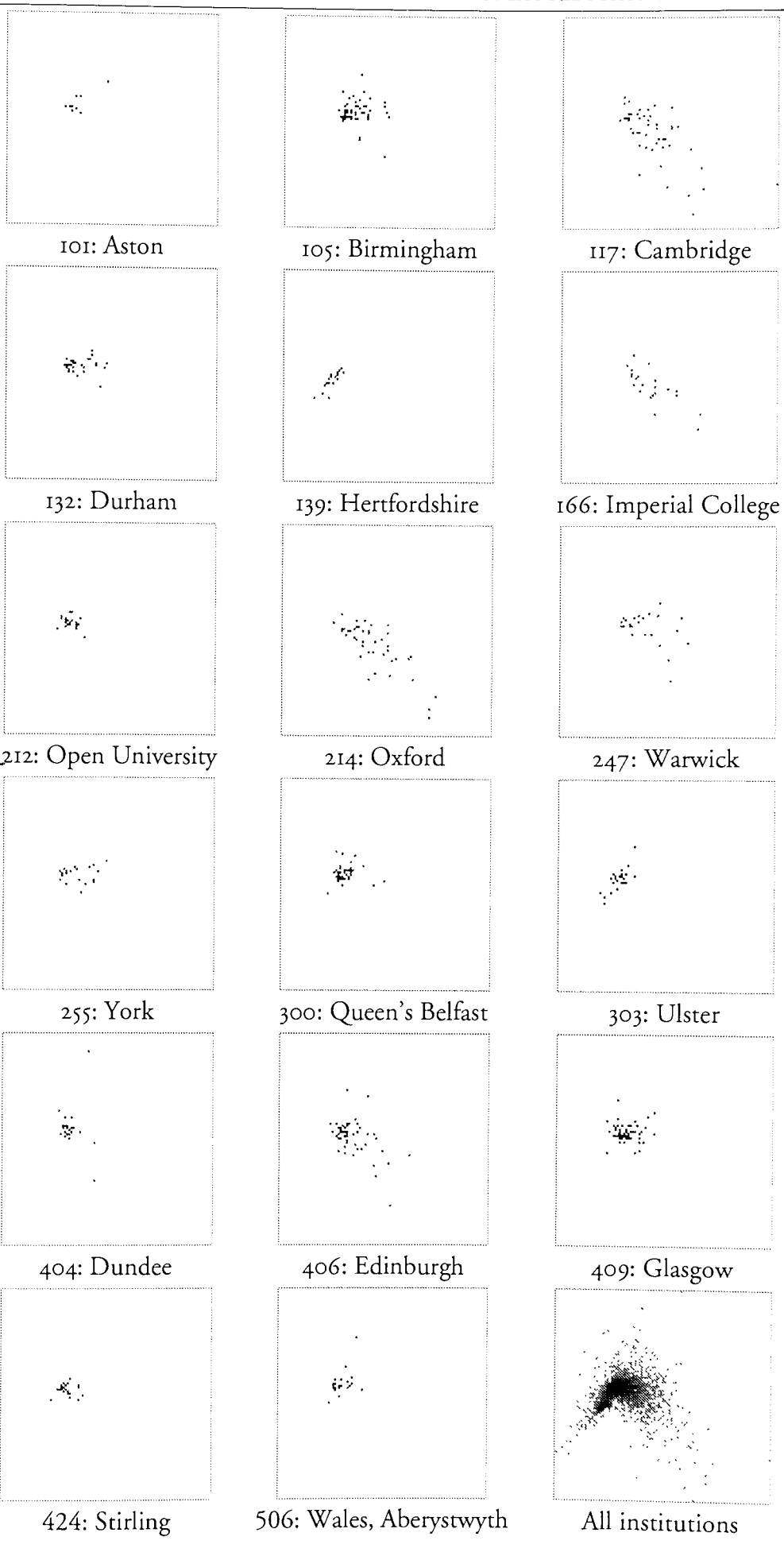


Figure 6.25 – RAE canonical variates, dependent upon institution

Figure 6.24 shows seventeen discipline plots which contain several examples characteristic of the majority of disciplines, together with a few exceptional plots. Some observations include:

- Most disciplines have a well-defined, fairly compact shape, suggesting that all the departments in the discipline share common features. French departments are particularly densely clustered.

- Hospital Clinical is by far the most anomalous discipline, as evidenced by the very dispersed canonical variates plot, and the frequent occurrence of Hospital Clinical departments as outliers in, for example, principal component projections.

- Engineering disciplines tend to occupy a different area of the plot than humanities. Chemistry and Physics are very similar, as are General and Civil Engineering.

- The Universities of Cambridge and Oxford are clear outliers in the Chemistry, Physics and General Engineering plots.

- Most of the departments in the left (low-rating) 'tail' are Business & Management

Figure 6.25 compares institutions in the same way. Again, characteristic and exceptional plots are shown, and some observations can be made:

- As with disciplines, most institutions occupy well-defined areas of the plot.

- Some institutions, including the University of Hertfordshire and the University of Ulster, have an elongated shape angled along the 'tail' of the plot; others, such as the Open University and the University of Glasgow, are more nebulous.

- Institutions which have many high ratings are more spread out – particularly the University of Cambridge, Imperial College and the University of Oxford.

6.8.3.5 Combined approach

Finally, a linear discriminant analysis was carried out on the database standardised by the first principal factor. The first two LDA axes are shown in figure 6.26. The axes are considerably less aligned to a small number of fields than the LDA axes from the raw database (seen in figure 6.22), and interestingly the PF_1 'size' factor itself is only of large significance in the second factor.

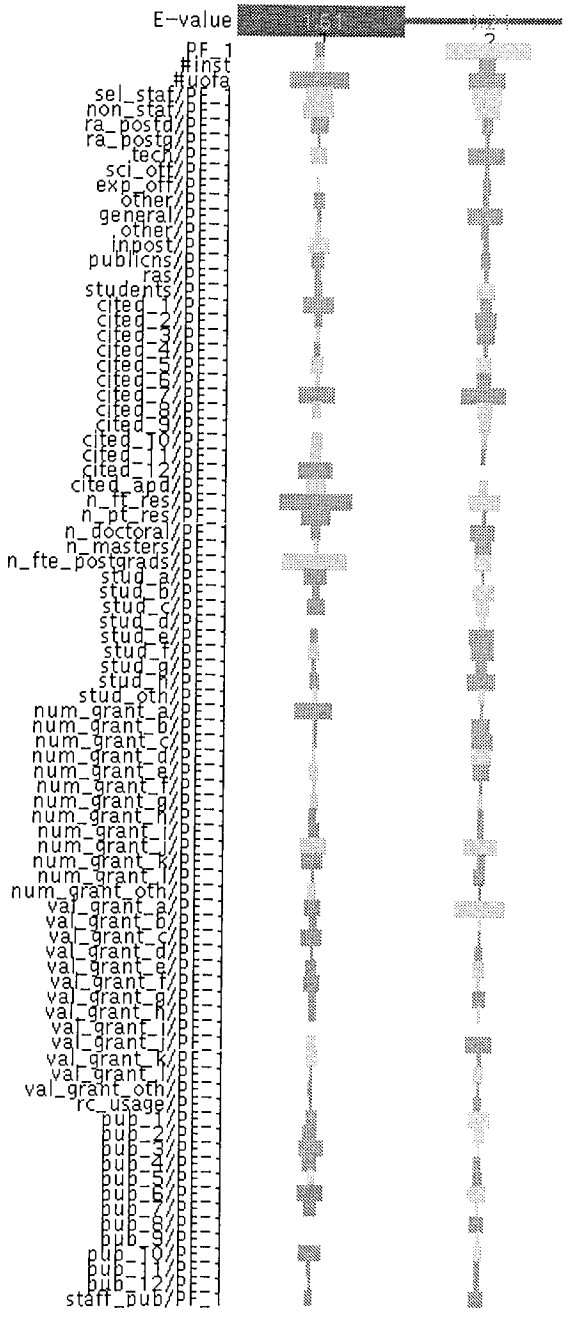


Figure 6.26 – Two most significant linear discriminant axes of the standardised RAE database

Plate 6.13 shows the first two canonical variates with `Rating` overlaid. Again, the plot is similar to that from the raw database (plate 6.12), but there are fewer outliers and the plot is generally 'cleaner'. It could well be considered the best 2-D view of the RAE database which shows that there is a clear structure to the ratings.

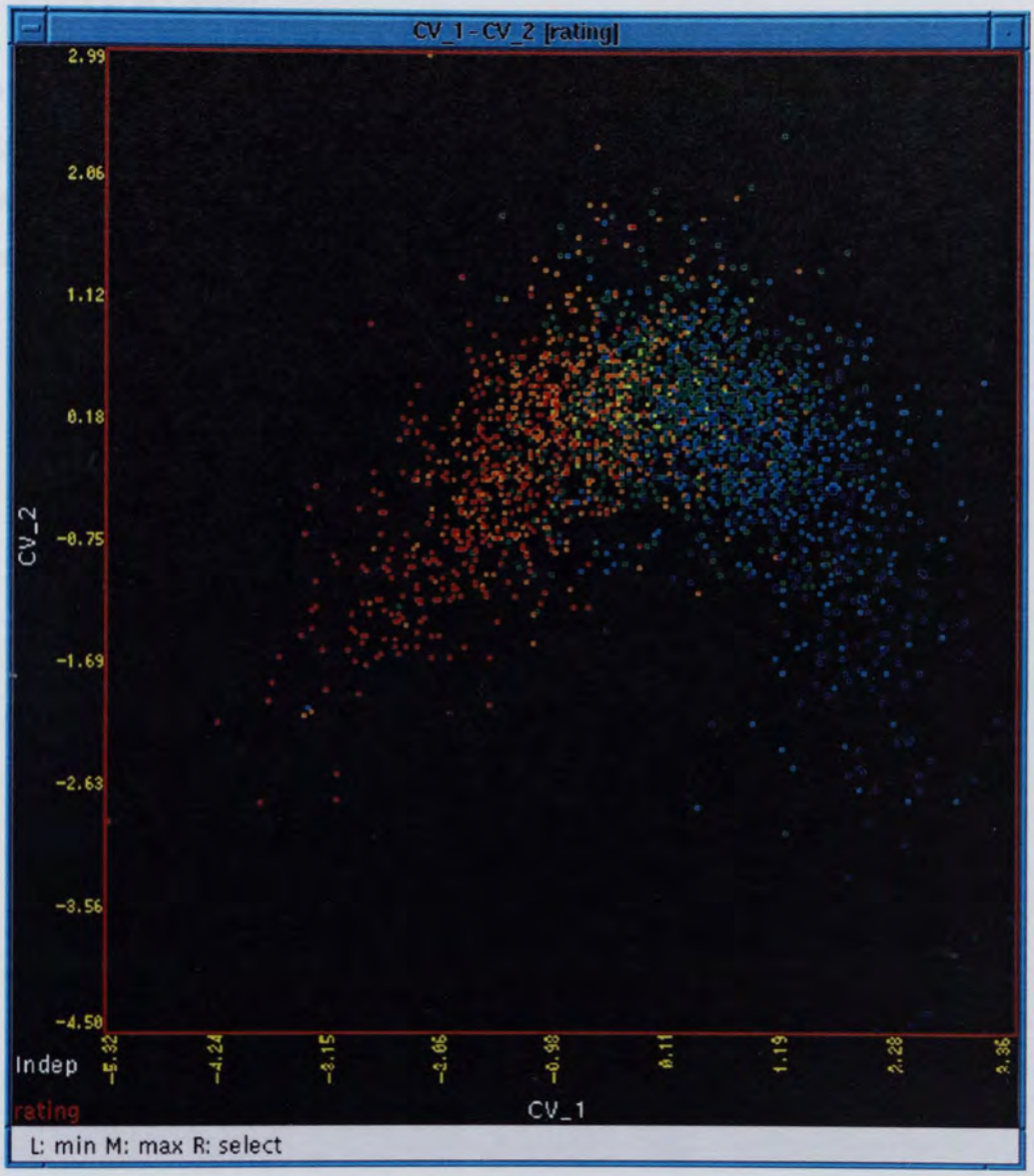


Plate 6.13 – Canonical variates of the standardised RAE database with `rating` overlaid

Again, the identity of some of the outlying and 'misplaced' departments in the plot were determined:

- The three red ones at the far left of the plot are Business & Management at Thames Valley, Nursing at Thames Valley and Business & Management at Salford.
- The orange two at the top is Computer Science at the University of the West of England.

- The green three amongst the red ones is Accountancy at Thames Valley; the green three amongst the blue fives is Hospital Clinical at Glasgow.
- The six blue fives at the bottom right of the plot are History, Physics and General Engineering at Cambridge and Physics, Chemistry and Hospital Clinical at Oxford.

Some of these departments were outliers in the canonical variate plot of the raw database; some were not. It definitely appears that Thames Valley is the 'worst' institution, Oxford and Cambridge are the 'best', Business & Management is the 'worst' discipline and Hospital Clinical the 'best'.

6.9 Conclusions

Linear dimensionality reduction techniques dramatically increase the data exploration power of the MADEN system.

The database can often be effectively 'summarised' by a small number of dimensions, which can usually be assigned easily-understood meanings, and observing the projection of the data onto the new dimensions can reveal clustering in the data which was previously hidden.

Principal component analysis is a fast method for reducing the number of dimensions, retaining as much information (in terms of variance) as possible in the new dimensions. However, its success with real data was generally confined to generating axes which could be interpreted to summarise the data, and the projections were not particularly revealing.

Principal factor analysis is a slower method, which requires the user to make the decision of how many factors to determine. In this implementation, however, the resulting factors are virtually identical to the principal component axes, which reduces the need for this technique. Factor rotation aids interpretation of axes, and can lead to more interesting projections of the data which clearly show how the data is clustered. The use of the MADEN divide operation was demonstrated by using the major factor of the RAE database to standardise the entire database, thereby enabling further investigation of its structure.

The new 'directed principal component analysis' technique proved to be of most use with the finance database, from which structure is otherwise difficult to extract. It is, however, rather too sensitive to binary fields which are well-correlated to the response field, and often generates dichotomised projections.

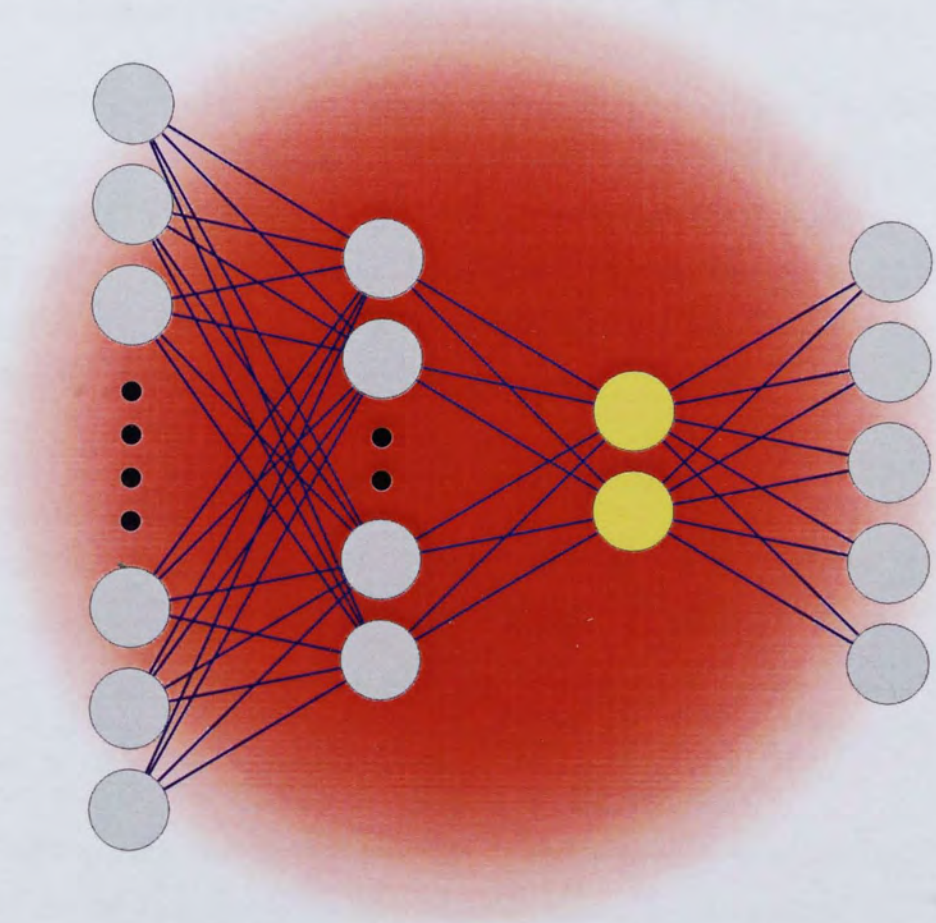
Projection pursuit, though shown to give excellent results on small test databases, is generally unsuitable for large 'real world' databases. The more powerful PP indices take an intolerably long time to optimise a pair of axes, and the simpler ones either take a fairly long time to generate (usually) uninteresting projections, or fail to converge to an optimum at all. Of the indices which performed well, the skew index gave by far the best projections of the data, particularly for the mail database.

Linear discriminant analysis is an impressive way to generate the best separation of classes which is possible using linear projection. It is fast and very powerful, and informs the user of how linearly separable the groups in the database are, both visually and numerically. With databases which only have two groups (e.g. the mail and finance databases), LDA is of limited use, as it can generate only one discriminant axis, which does not generally reveal much of the database's hidden structure.

Remarkable results were obtained from the RAE database using linear dimensionality reduction, particularly LDA: These will be discussed further in chapter 8.

The next chapter moves on to consider non-linear dimensionality reduction techniques, which (in theory, at least) offer even more powerful methods for data analysis.

Chapter 7



Dimensionality Reduction II: Non-linear Methods

The statistical likelihood is that other civilizations will arise. There will one day be lemon soaked paper napkins. Till then, there will be a short delay. Please return to your seats.

[Adams, 1980]

7.1 Introduction

All the dimensionality reduction methods considered so far have consisted of determining a set of axes through the data space and linearly projecting the data onto them. This chapter describes some non-linear techniques and their application to dimensionality reduction, in an attempt to extract more information than is possible with linear methods, or maybe different information altogether.

7.2 Traditional Methods

7.2.1 Sammon's non-linear mapping

The non-linear mapping (NLM) [Sammon, 1969] is a point mapping from the data n-space into a lower-dimensional k-space such that the interpoint distances in the k-space approximate the corresponding interpoint distances in the n-space.

Mathematically, let d_{ij}^* be the distance (measured by a suitable metric) between points i and j in the n-space, and d_{ij} be the distance between the mapped positions of the same points in the k-space. The NLM algorithm initially positions the mapped points at the projection of the data onto the first k PCA axes, then optimises the positions of the mapped points so as to minimise the error measure (sometimes known as the 'stress') E, as defined by equation 7.1.

$$E = \frac{\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{\left[d_{ij}^* - d_{ij}\right]^2}{d_{ij}^*}}{\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} d_{ij}^*} \qquad 7.1$$

The technique has two serious drawbacks:

- It requires $O(n^2)$ distance calculations to be performed merely to calculate E, and the same number of distances d_{ij}^* to be stored. For large databases this is wholly impractical.

- Each point is positioned to optimise E – following optimisation, there is no 'rule' for mapping points from n-space into k-space. This precludes generating a Sammon mapping using a representative sample of points and then mapping the entire database.

These problems make the Sammon mapping inappropriate for use in MADEN.

7.2.1.1 Application of neural networks

Attempts have been made to use neural networks to perform a transformation similar to the Sammon mapping [Tattersall & Limb, 1994], and recent work in the Aston Neural Computing Research Group [Lowe & Tipping, 1995] has produced a neural network called NEUROSCALE which can also incorporate 'subjective dissimilarity'. This modifies d_{ij}^{*} to include, for example, class information, which forces points drawn from different classes to be further apart in the k-space than they are in n-space.

Unfortunately, NEUROSCALE still requires the $O(n^2)$ calculations and distance storage of the original NLM algorithm. The development of NEUROSCALE is continuing, but was insufficiently advanced at the time of the work described in this chapter to be of practical application in MADEN.

7.2.2 Multidimensional scaling

Multidimensional scaling (MDS) [Mardia *et al*, 1979; Davison, 1983] is a term which encompasses a large number of techniques, all of which take as input the matrix of scalar distances or dissimilarities between each pair of data points and generate a low-dimensional representation of the original data which preserves as much of the distance information as possible. Once again, though, the need for a full dissimilarity matrix rules out the use of MDS in MADEN.

7.2.3 Principal curves and surfaces

Principal curves and surfaces (PC/S) [Hastie & Stuetzle, 1989; Malthouse, 1995] is a non-linear version of PCA which uses a set of curves or surfaces in the data space, and projects each data point to the point on the curve or surface to which it is closest. However, the PC/S algorithm is highly complex and at present is only available in the *S* software.

7.2.4 Conclusions

The traditional methods of non-linear dimensionality reduction are unsuited to application to the databases under investigation in MADEN, even when approached from the perspective of neural networks.

There are several alternative applications of neural network techniques, however, which are easily applied to the task, as will be shown in the remainder of this chapter.

7.3 Kohonen Self-organising Map Revisited

7.3.1 Concept

The Kohonen self-organising map was introduced in chapter 5 for the purpose of clustering the data by using the weight vectors of the Kohonen map as cluster centres.

The map also lends itself quite naturally to dimensionality reduction: each data point is applied to the map, and the map coordinates of the 'winning node' are used as the coordinates of the point in a new two-dimensional space.

7.3.2 Implementation

The Kohonen clustering routine in MADEN was augmented to produce two new overviews following training: the weight vector overview previously described, and a new 'map' overview containing the original database with the two additional coordinate fields. In the same way as the coordinate fields in the weight vector overview, the x coordinate is doubled and offset by one if the y coordinate is odd.

An enlargement of the density plot of the two coordinate fields (Kohonen_x and Kohonen_y) can then be created. Immediately the density of the data on the map becomes apparent, shown by the size of the rectangles at each node – as shown in figure 7.1 overleaf. Using this enlargement window, any of the fields in the original database may be overlaid to see how they vary across the map. In particular, the response field may be used as an overlay, to investigate whether the map has segregated the data into the groups in which we are interested.

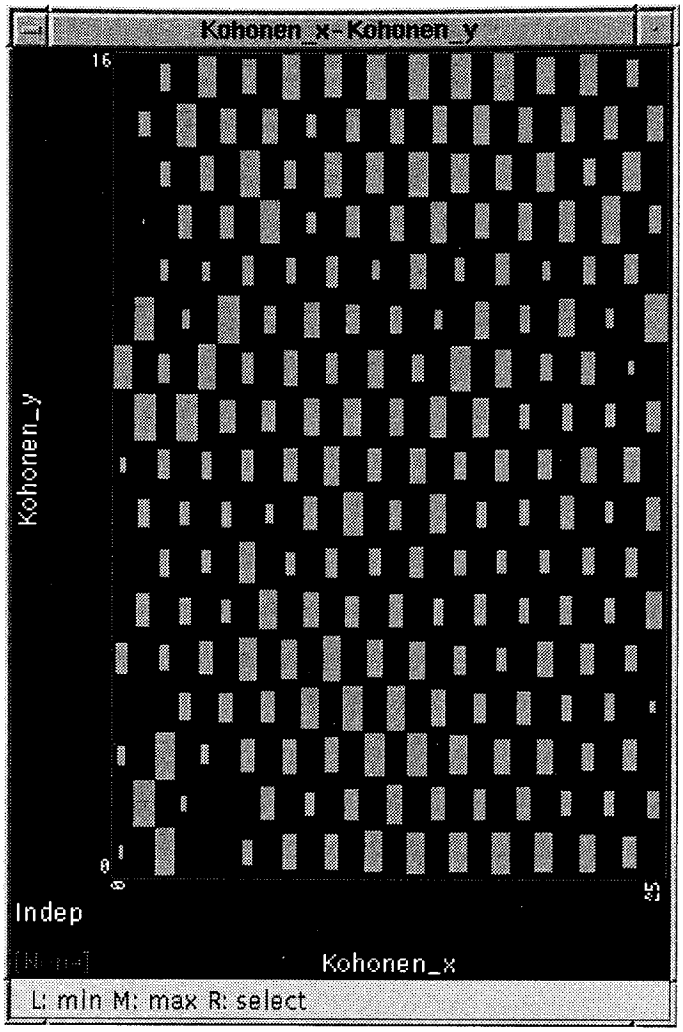


Figure 7.1 – Enlargement of the Kohonen map coordinates showing data density

7.4 Multi-layer Perceptron Hidden Layer

7.4.1 Introduction to MLPs

A multi-layer perceptron (MLP) neural network is comprised of a number of 'layers' of 'neurons'. The first layer, known as the input layer, has one node for each scalar component of the input vector (in this case, one per input field such as `Age`, `Sex:M` etc). Each neuron has a scalar 'activation'. The activation of a neuron in the input layer is simply the value of its input; the activation of each non-input neuron is a function of the weighted sum of the activations of all the neurons in the previous layer.

Mathematically, if O_i is the activation (output) of neuron i, then

$$O_i = \mathrm{f}\left(\sum_{j \in L_i} W_{ij} O_j\right) \qquad 7.2$$

where W_{ij} is the weight which determines how much effect the jth neuron has on the ith neuron, L_i is the set of neurons in the layer before neuron i, and f is the activation function.

7.4.1.1 Choice of activation function

For optimisation purposes, it is desirable that the activation function be differentiable. Typically the activation function is chosen so that it is sigmoidal in shape, i.e. it asymptotically approaches saturation values for large positive and large negative inputs. Standard activation functions (and their derivatives) are shown in equation 7.3:

$$\begin{array}{lll} \text{sigmoid} & \mathrm{f}(x) = \dfrac{1}{1+\mathrm{e}^{-x}} & \mathrm{f}'(x) = \mathrm{f}(x)\big(1-\mathrm{f}(x)\big) \\ \tanh & \mathrm{f}(x) = \tanh(x) = \dfrac{\mathrm{e}^{x}-\mathrm{e}^{-x}}{\mathrm{e}^{x}+\mathrm{e}^{-x}} & \mathrm{f}'(x) = 1-\mathrm{f}(x)^2 \\ \text{linear} & \mathrm{f}(x) = x & \mathrm{f}'(x) = 1 \end{array} \qquad 7.3$$

7.4.2 Application of an MLP to dimensionality reduction

An MLP can be trained to predict the response field of the database from the remaining fields. If the network architecture contains a 'bottleneck', the activations of the nodes in the bottleneck can be used as the new dimensions of a lower-dimensional representation of the data. The bottleneck nodes perform a kind of non-linear discriminant analysis, since they are trained to contain as much information as is necessary to successfully predict the response field. This approach has recently been presented by Mao and Jain, who refer to it as NDA (for non-linear discriminant analysis) [Mao & Jain, 1995].

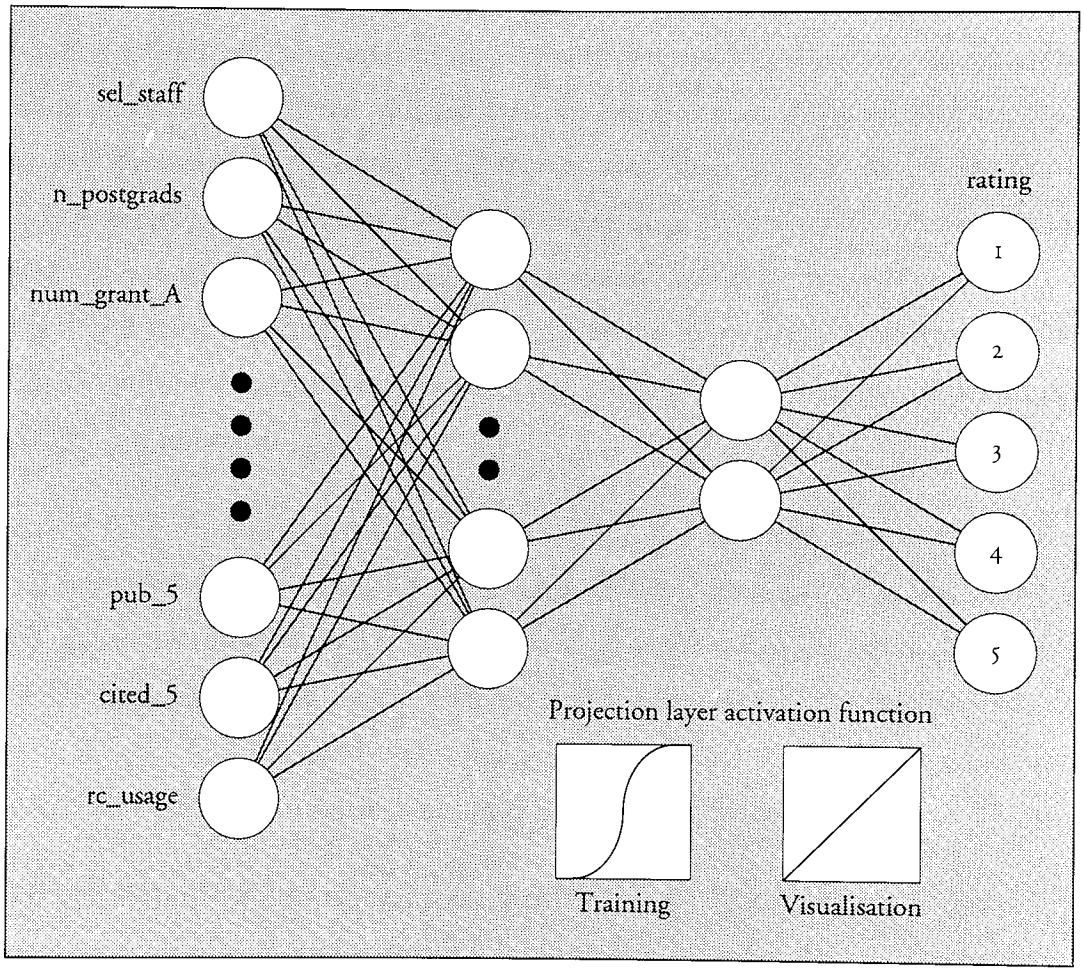


Figure 7.2 – Multi-layer perceptron architecture (for RAE database)

The network architecture is shown in figure 7.2. There is one input node (on the left) for each non-response field in the database, and one output node for each discrete response. The figure shows the network for the RAE database, so there are five output nodes. Between the input and output layers are two hidden layers, the second (the 'projection layer') having one node for each dimension required in the reduced space.

Two hidden layers are required to allow a general non-linear function between the

inputs and the projection layer; without the first hidden layer, the projection layer activations would be a linear combination of the inputs.

The activations of the bottleneck nodes of a trained network (which can correctly predict the response field) will tend to saturate. Visualising these activations generates a plot with the majority of points lying in the corners. Following Mao, the tanh activation functions for the last hidden layer (the projection layer) are replaced by a linear function in order to generate the projection, once training is complete.

7.4.3 Implementation

Neural network code developed in the Aston University Neural Computing Research Group was easily integrated into MADEN. An 'MLP' menu was placed at the top of the overview window, offering the choice of two, three or four dimensions in the projection layer. When the user chooses one of these menu items, the standardised, category-expanded database is passed to the neural network code, which extracts the target field (the response). An MLP network of the appropriate architecture is then created, with tanh activation functions, and trained by means of the same CG code used for projection pursuit.

During training, the network error is displayed, to allow the user to monitor the training process, which takes several minutes. When training is complete (when the training error cannot be further reduced, or 25 passes through the database have been made), the predicted value of the response field (i.e. the label of the most-active target node) for each input record is calculated and stored. A confusion matrix is displayed, to indicate how well the network has learned to predict the target.

Another network is now constructed, identical to the first, except that the original target layer is omitted, and the activation functions in the projection layer are made linear. The inter-node connection weights from the original network are copied into the new network, each record in the database is presented in turn, and the activations of the projection layer nodes are stored.

Finally, a new overview is created, displaying the original database together with additional fields containing the predicted output and the (linear) activations of the projection layer nodes.

7.5 Multi-layer Perceptron Autoencoder

7.5.1 Introduction

An alternative application of the MLP is to construct a network architecture which has the same target as its input, with a bottleneck in between. The network must then learn how to reconstruct the input vector using only the information distilled into the bottleneck nodes. Hence the activations of the bottleneck nodes can be seen as performing a non-linear principal components analysis (NLPCA) [Kramer, 1991], also known as 'principal manifolds' [DeMers & Cottrell, 1993]. Because the bottleneck layer automatically encodes as much information about the input vector as possible within the bottleneck nodes, this type of network can be described as an *autoencoder*.

7.5.2 Architecture

Figure 7.3 shows the architecture of the autoencoder network used in MADEN.

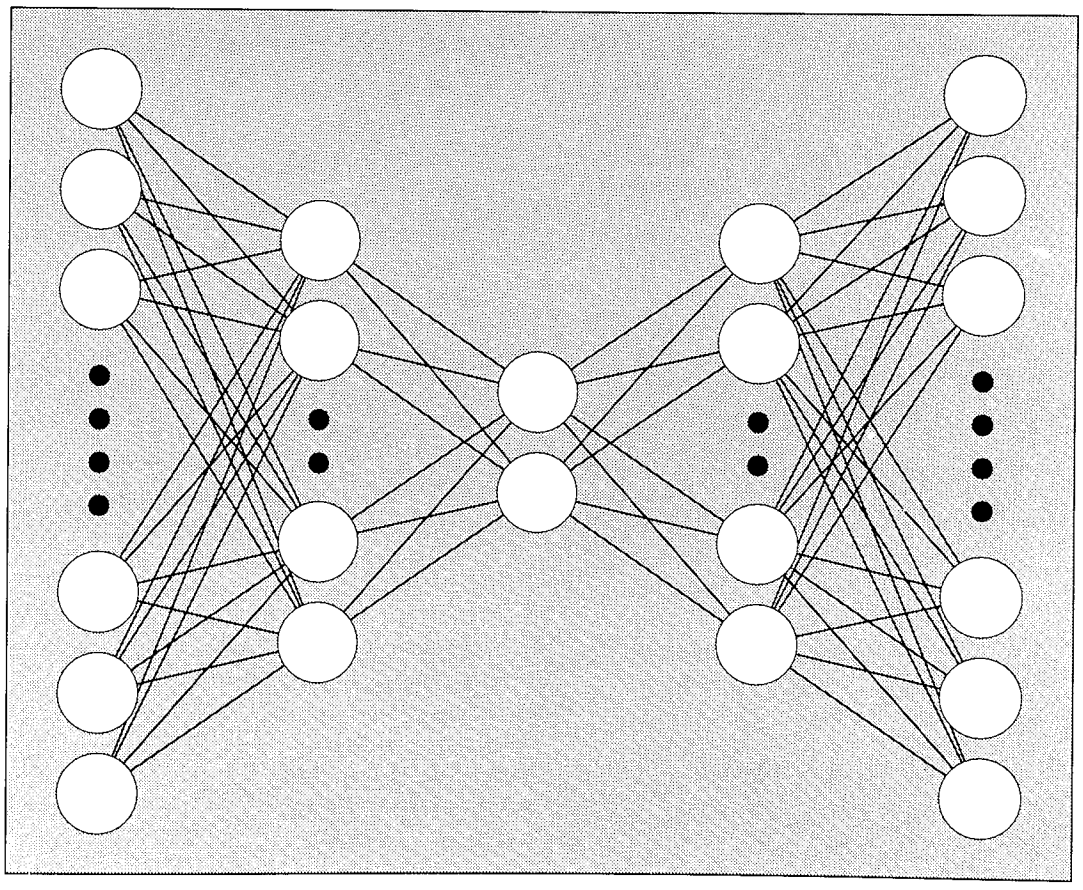


Figure 7.3 – Autoencoder architecture

Five layers are required in order that the transforms from the input vector to the bottleneck and from the bottleneck to the target vector can be general non-linear functions.

7.5.3 Implementation

Implementing the autoencoder was relatively straightforward, since it is virtually identical to the previous method.

After the user chooses the size of the projection layer from a menu, the neural network is constructed and then trained using the CG method. Training is necessarily slower than for the previous technique, due to the larger number of nodes and weights.

Following the completion of training, a new network is created with only the first three layers of the five-layer network, and the weights are copied across. The projection layer activation functions are made linear and their activations for each data record are recorded. Then a new overview is created, containing the original data together with the activations of the projection layer nodes.

7.6 Use with Real Data

7.6.1 Mail database

7.6.1.1 Kohonen mapping

A Kohonen net, identical to the one used to demonstrate clustering, was trained on the mail database, and an enlargement of the Kohonen_x-Kohonen_y plot was opened and resized to show the hexagonal grid.

Plates 7.1 and 7.2 on the following pages show this enlargement with eight different overlays. The number of data records mapped to each node is indicated by the size of the node's rectangle.

Because the overlay fields are fields of the actual data the colours of the nodes represent the average value of the overlay field for the records mapped to that node, rather than the respective component of the weight vectors used previously. The overlay of an actual data field will strongly resemble that of the weight vector component, but will not necessarily be identical to it.

Plates 7.1*a*, *b* and *c* overlay the fields Ac_Turn, Minbal and Maxbal, the same fields taken from the weight vectors in plates 5.1*a*, *c* and *d*, and indeed the similarity is clear. This indicates that the Kohonen mapping has trained well, so that each node's weight vector lies near the mean of the data records which are assigned to it.

Plate 7.1*d* uses Dcard as an overlay, and demonstrates that there is not an obvious variation of this variable across the map, though there are nodes with larger and smaller than average proportions of debit card holders, and maybe generally fewer towards the top half of the map.

Plate 7.2*a* examines the variation of Marital:M. The purple colour of most nodes is the overall average value of this field. This should be contrasted with plate 5.2*a*, which used the weight vector component Marital:S. The three areas of non-married customers correspond neatly to the nodes which are active for single customers.

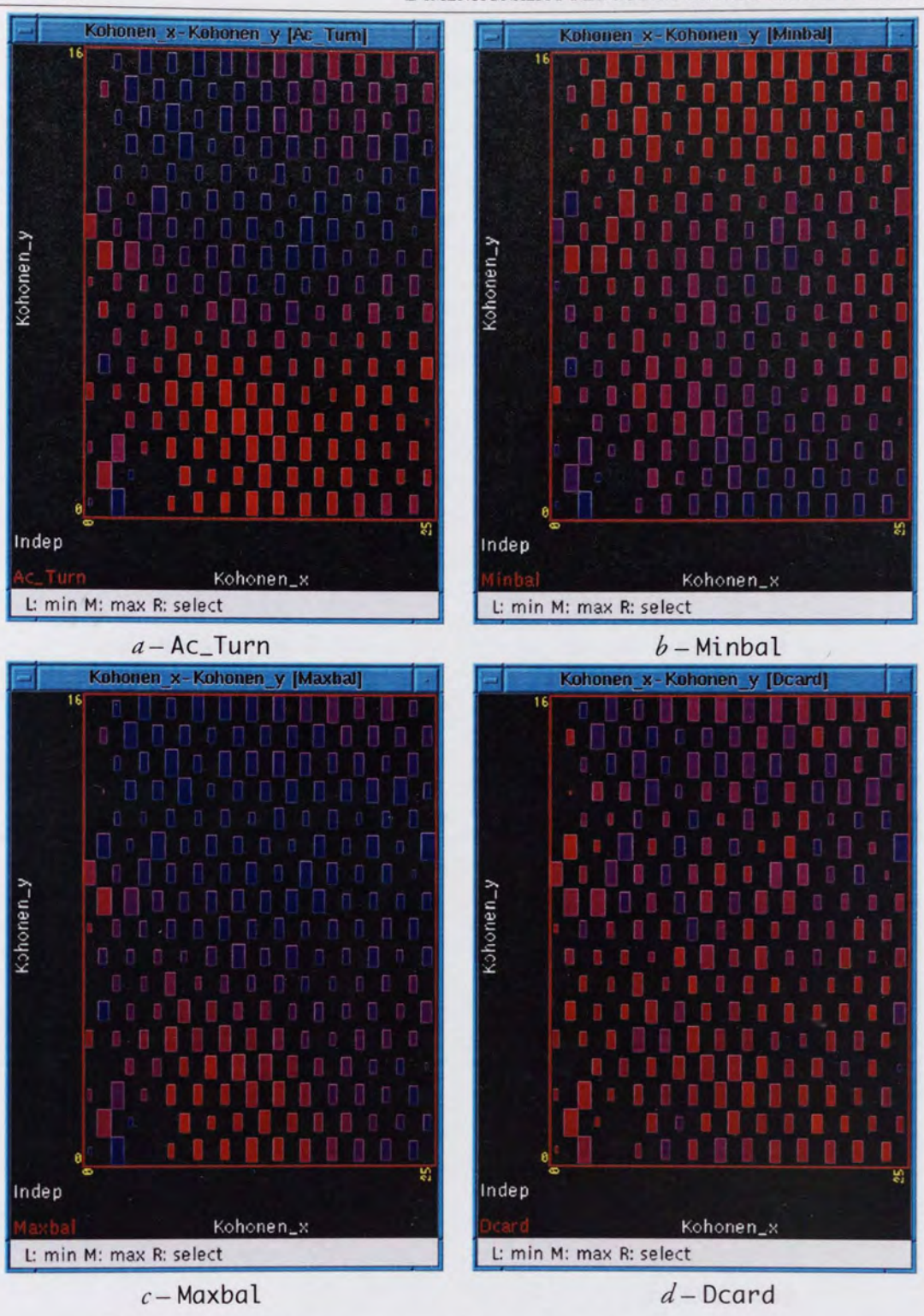
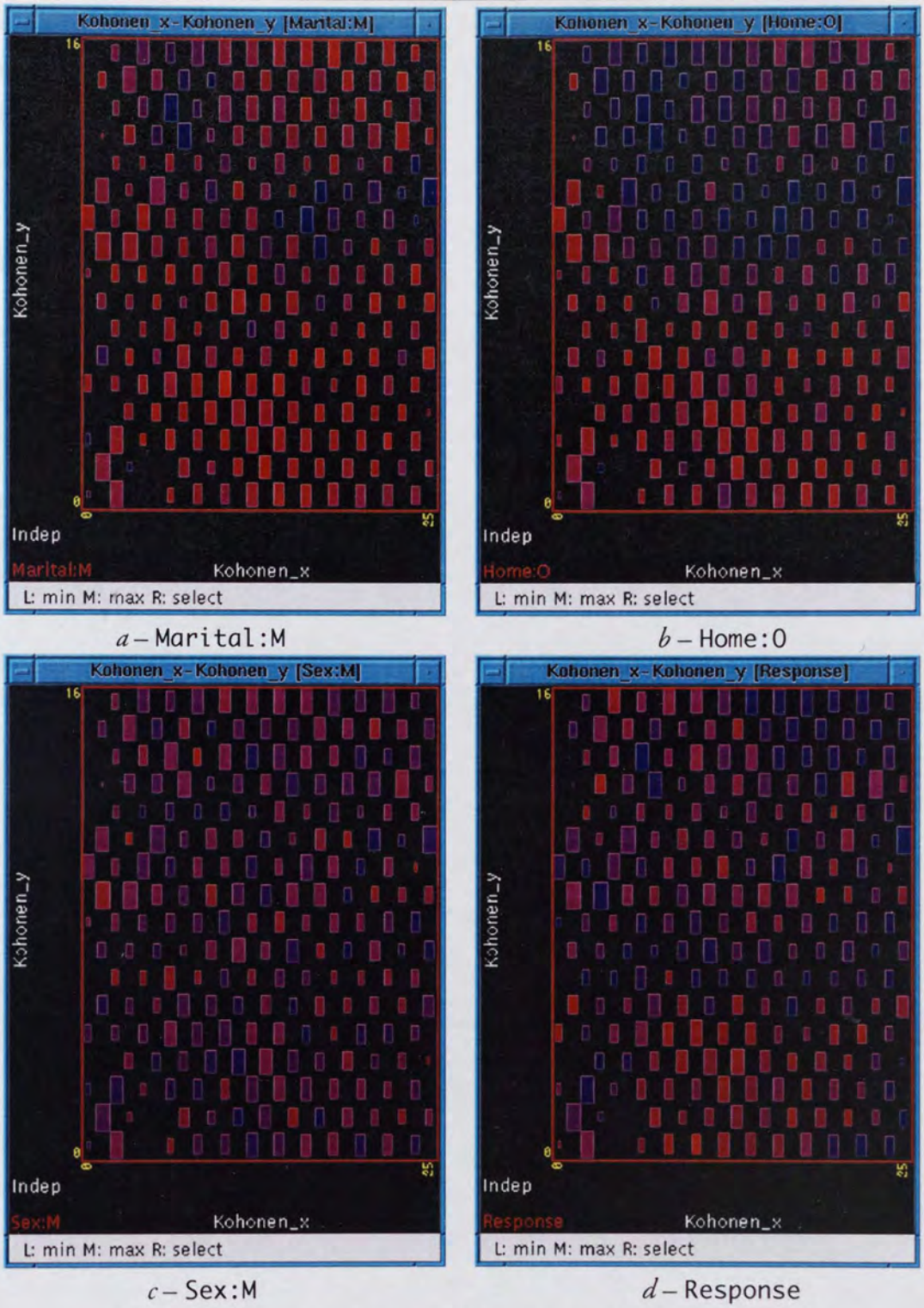


Plate 7.1 – Kohonen mapping of the mail database with overlays

Plate 7.2*b* looks at the `Home:O` field, and is visibly correlated with the `Marital:M` overlay shown in plate 7.2*a*. As previously noted, unmarried customers tend not to own their own homes. Beyond this similarity, a general pattern of home ownership can be seen across the plot, with some nodes having a large proportion of home owners, and some having very few.



a – Marital:M *b* – Home:0

c – Sex:M *d* – Response

Plate 7.2 – Kohonen mapping of the mail database with overlays

Plate 7.2*c* reveals that the Kohonen map has not separated the sexes to any great extent. Most nodes are uniformly purple, indicating roughly equal proportions of male and non-male (i.e. female and unknown) customers.

Plate 7.2*d* uses Response as its overlay, which is not possible when using only the weight vectors. It shows the location of responding customers following the Kohonen dimensionality-reducing transformation. As might be expected, there is no dramatic

dividing line between red and blue, but some areas can be discerned: areas of low response are located in the top right corner, across the middle, and in the bottom right corner of the map. Higher levels of response can be seen on three individual nodes, and generally towards the bottom middle of the map.

7.6.1.2 Hidden layer

Plate 7.3 below shows the result of training an MLP with a three-node bottleneck on the mail database. The activations of the three hidden nodes are shown, with the predicted response (the output of the network) overlaid.

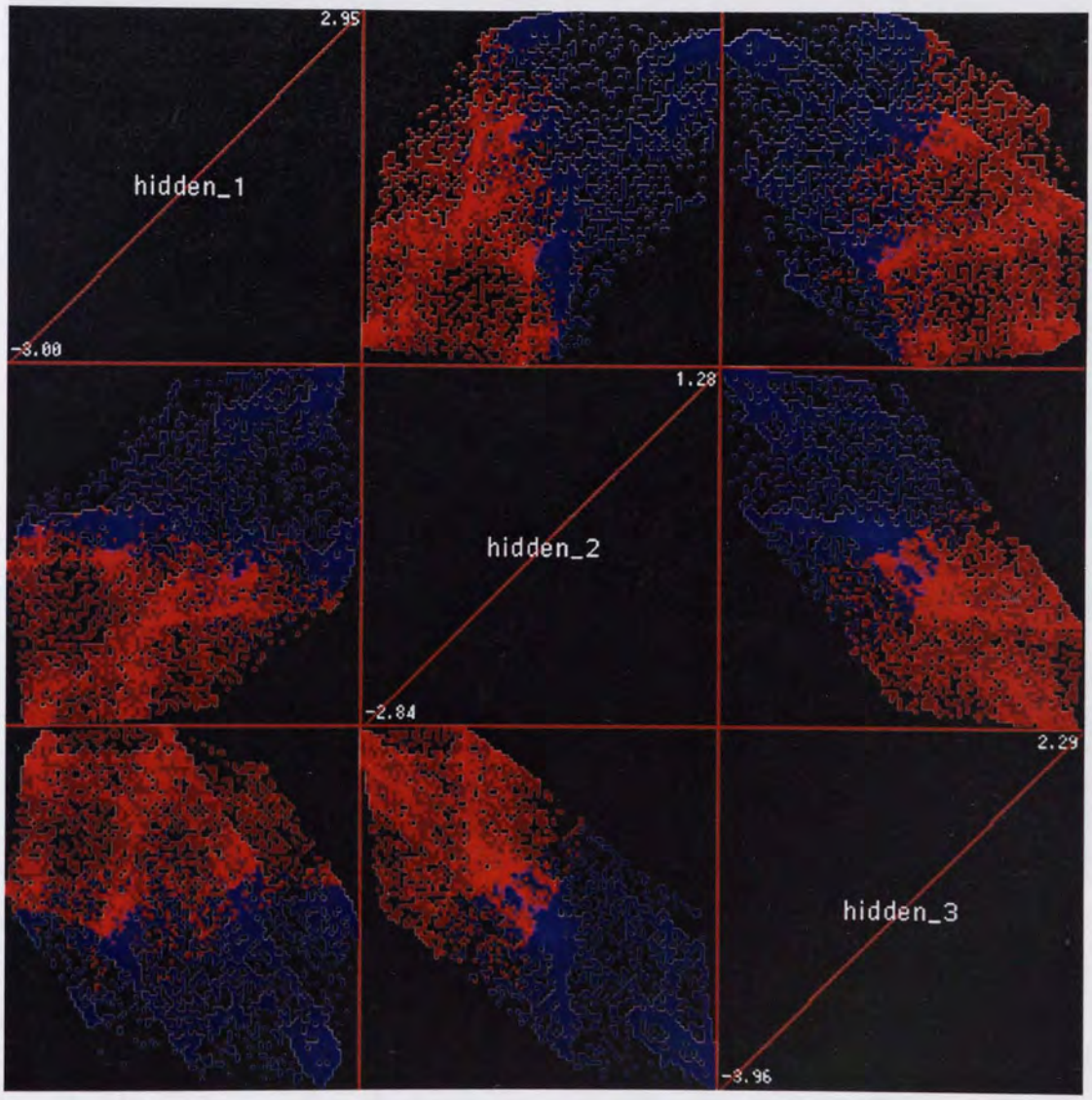


Plate 7.3 – Results of training a three-node bottleneck MLP on the mail database, with predicted response overlaid

Clearly the MLP has created a complex three-dimensional representation of the high-dimensional database. Sub-structures can be seen within the 'data cloud', particularly

in the hidden_3–hidden_1 plot, and at the top right of the hidden_2–hidden_1 plot.

The coloured overlay shows the location of the output neuron's hyperplane which defines the boundary between responders and non-responders. This should be compared with plate 7.4 below, which uses the actual Response field as its overlay.

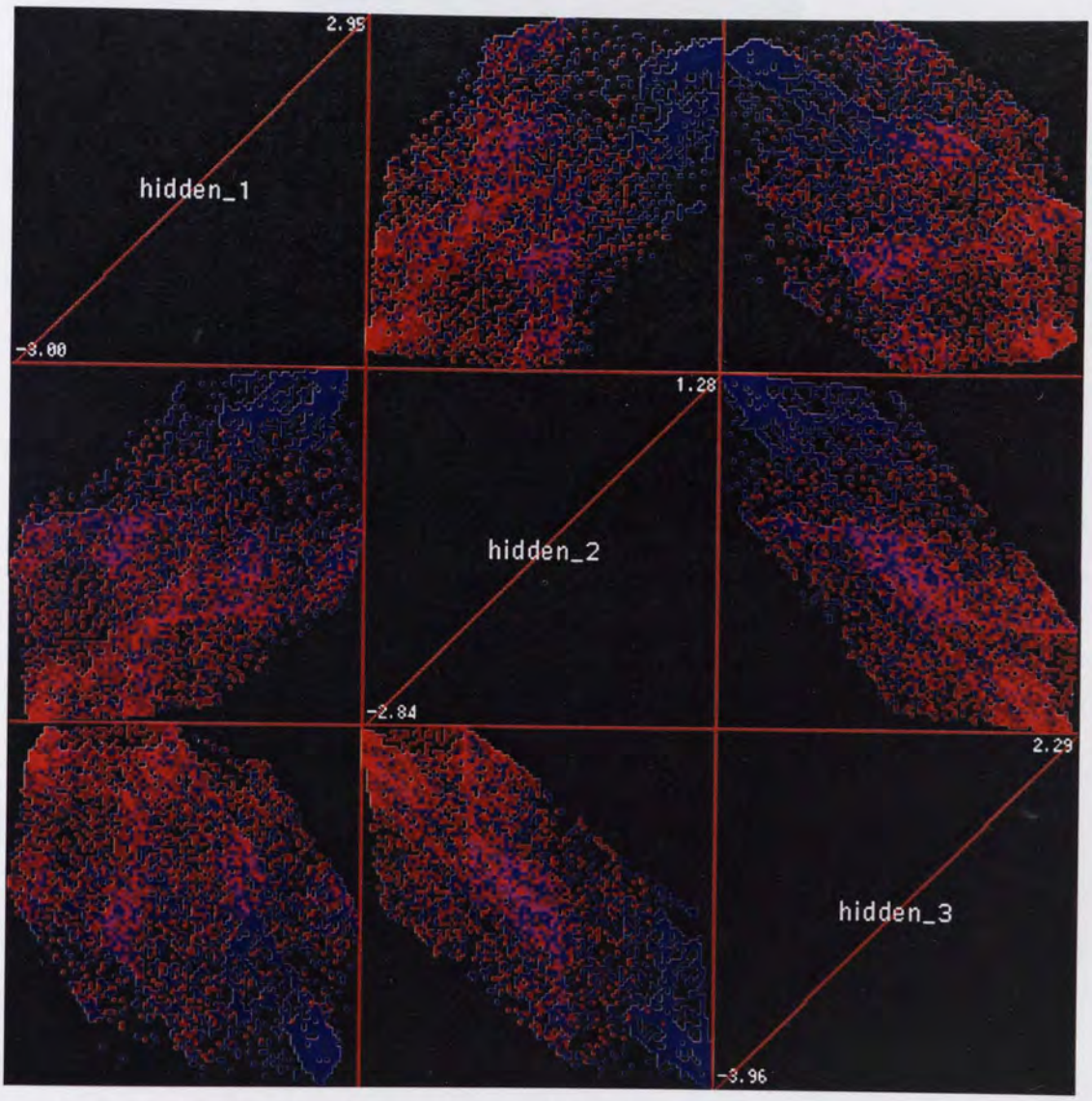


Plate 7.4 – Results of training a three-node bottleneck MLP on the mail database, with actual Response overlaid

The separation and clarity of the red and blue areas in plate 7.4 shows how well the MLP has learned to predict whether a customer will respond or not. It seems that the network has been fairly successful in its task, as there are clear areas of blue and red, though the interface is somewhat wide and confused.

In fact, by examining the plot of predicted against Response and using selection, we discover that 68.8% of customers were correctly predicted.

As a final example of how the MLP has separated the two groups, a linear discriminant analysis was performed on the post-MLP database, clipped to contain only the three hidden layer fields and Response. The separation measure F was 1.32, indicating a reasonable separation, as demonstrated by the plot of the resulting canonical variate shown in plate 7.5.

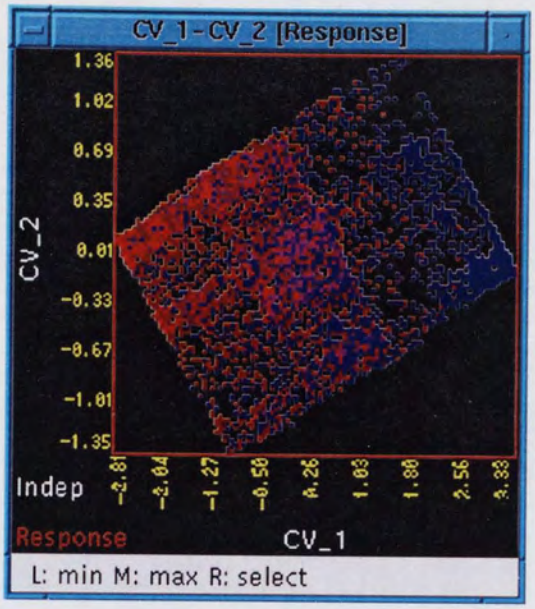


Plate 7.5 – Canonical variate of the hidden layer activations, with Response overlaid

7.6.1.3 Autoencoder

A three-node autoencoder was then trained on the mail database, resulting in the plots shown in figure 7.4.

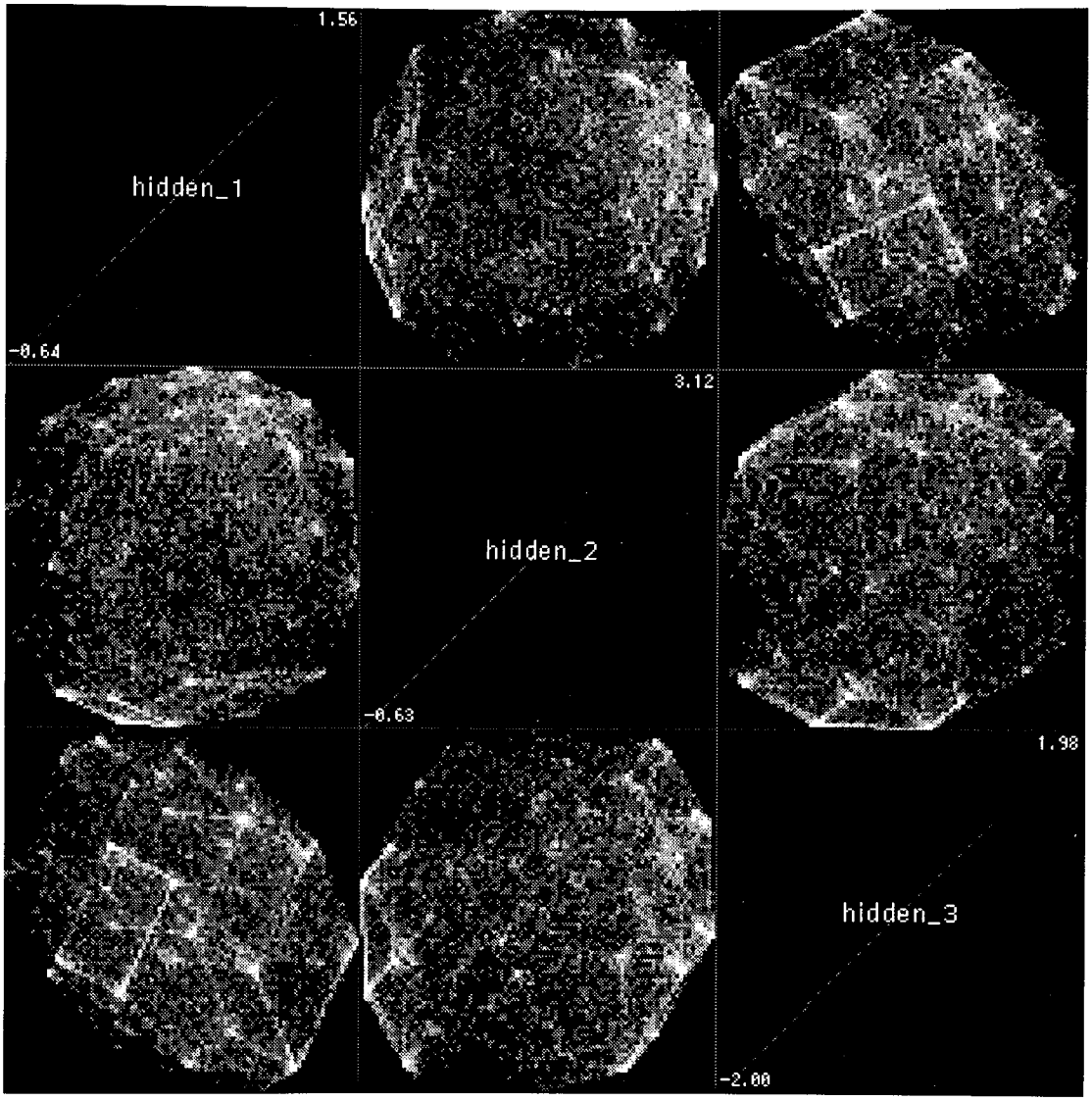


Figure 7.4 – Three-dimensional autoencoding of the mail database

As with the network trained to predict the response field, the activations of the bottleneck nodes of the network trained to reconstruct its input show a large amount of internal structure. Several rectangular areas can be seen in the three-dimensional space created by the hidden_n fields.

This network has also made more uniform use of the 3-D space available to it than the MLP used in the previous section did, resulting in a 3-D representation which is nearly spherical. The information required to reconstruct a complete vector of customer details evidently requires more of the space to be used.

Plate 7.6 below shows the same overview of the three autoencoder fields, with Response overlaid. There is no clear pattern of responders in this three-dimensional representation of the database, confirming once again that there is no simple rule for predicting customer response given the remaining customer data.

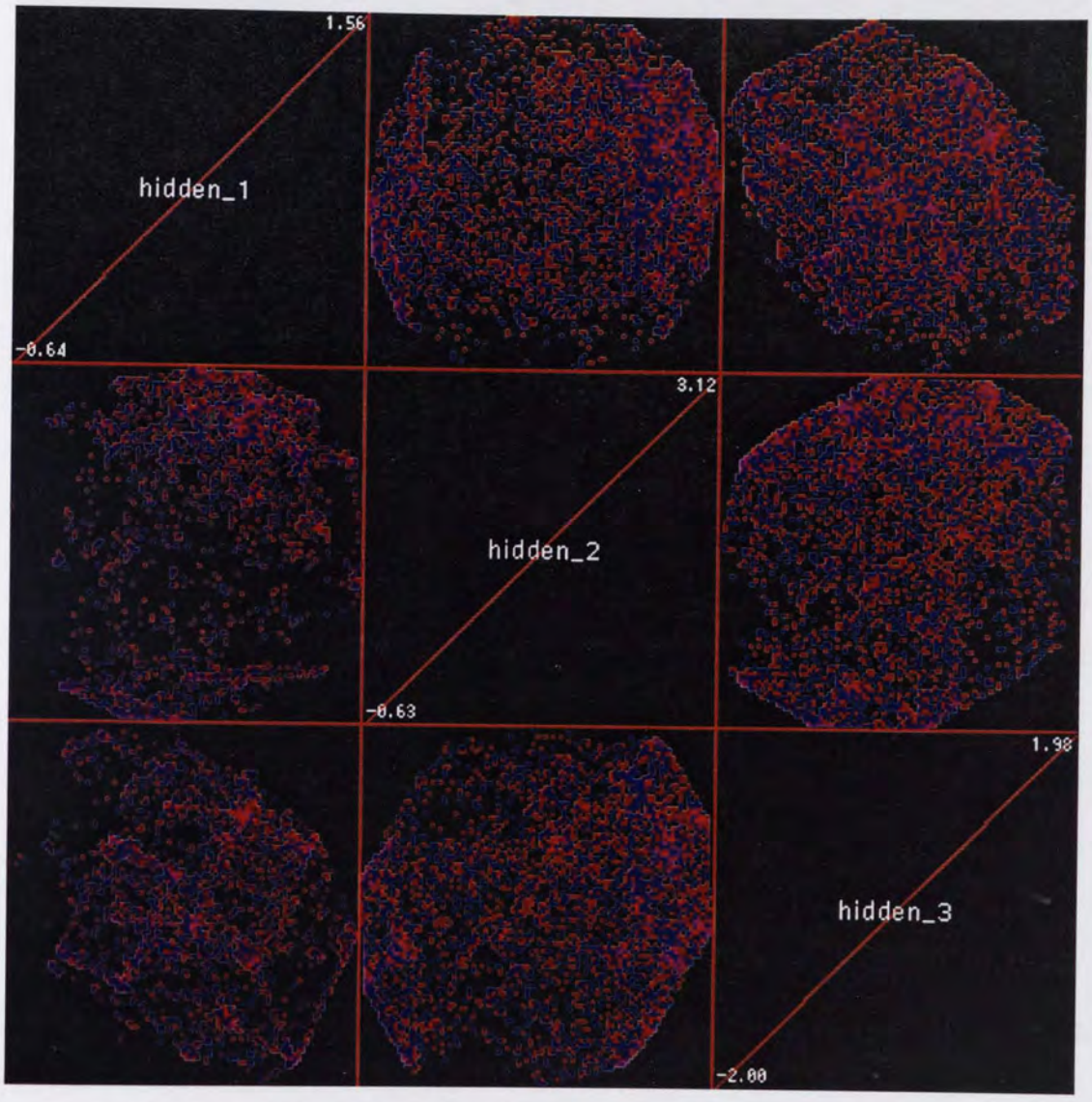


Plate 7.6 – Three-dimensional autoencoding of the mail database, with Response overlaid

The following three pages show the autoencoding of the mail database, with three of the original fields overlaid, to investigate which areas of the low-dimensional space are used to code which features of the high-dimensional data space.

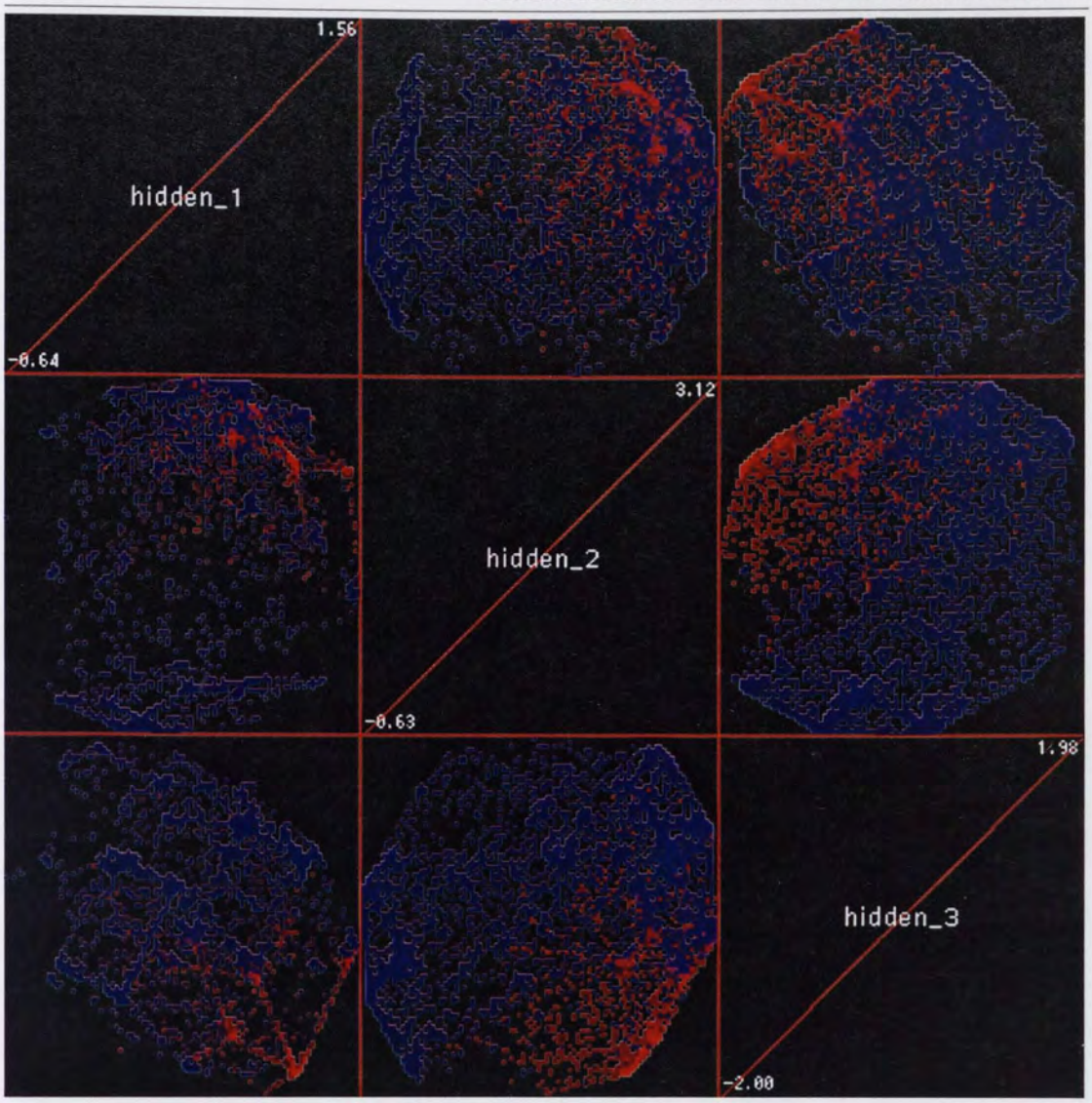


Plate 7.7 – Three-dimensional autoencoding of the mail database, with Home:R overlaid

Plate 7.7 shows where the customers who rent their homes are mapped to: a small well-defined area of the three-space.

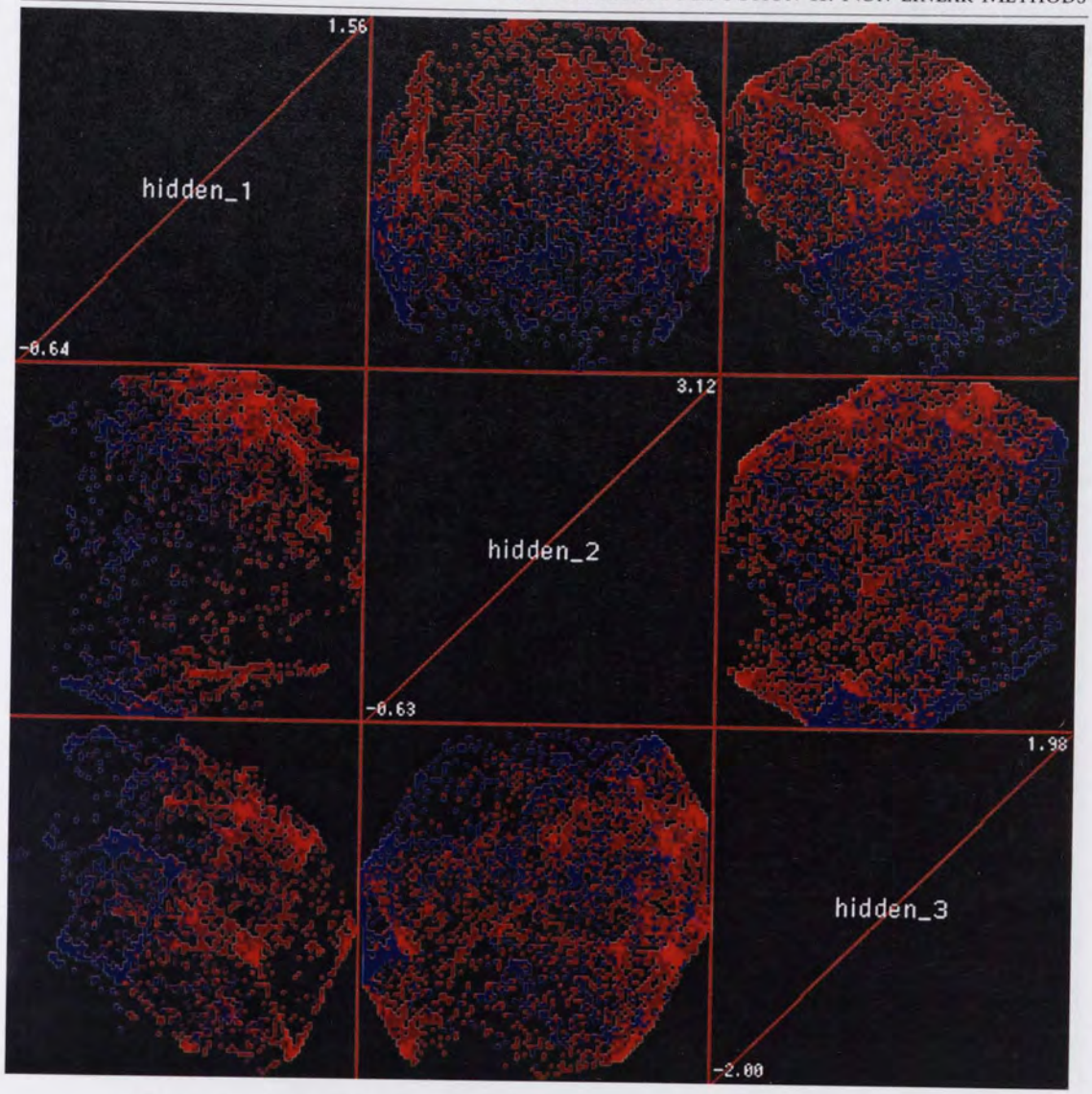


Plate 7.8 – Three-dimensional autoencoding of the mail database, with `Marital:M` overlaid

Plate 7.8 locates the married customers: a larger area of the space, including most of the area in which the home-renting customers are placed.

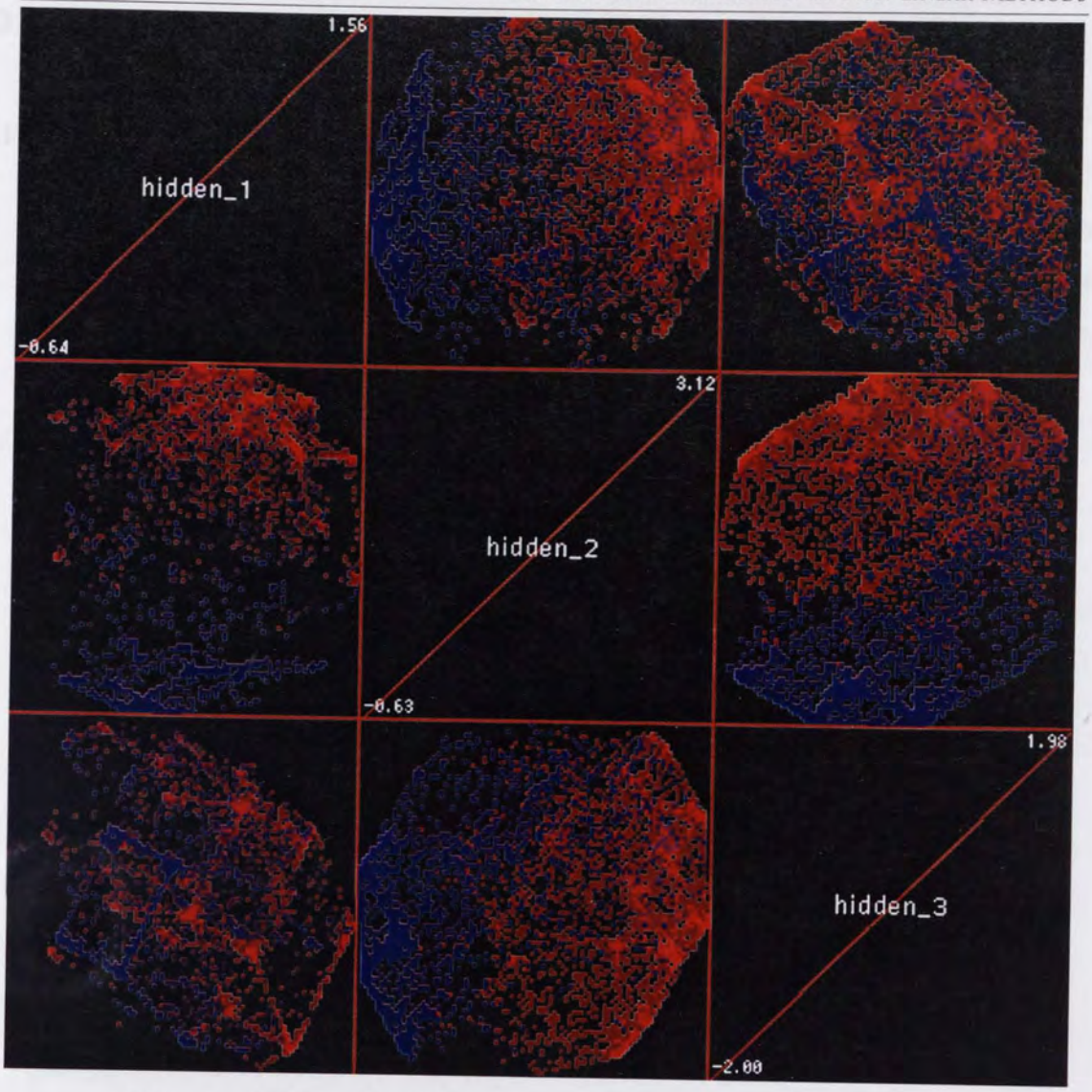


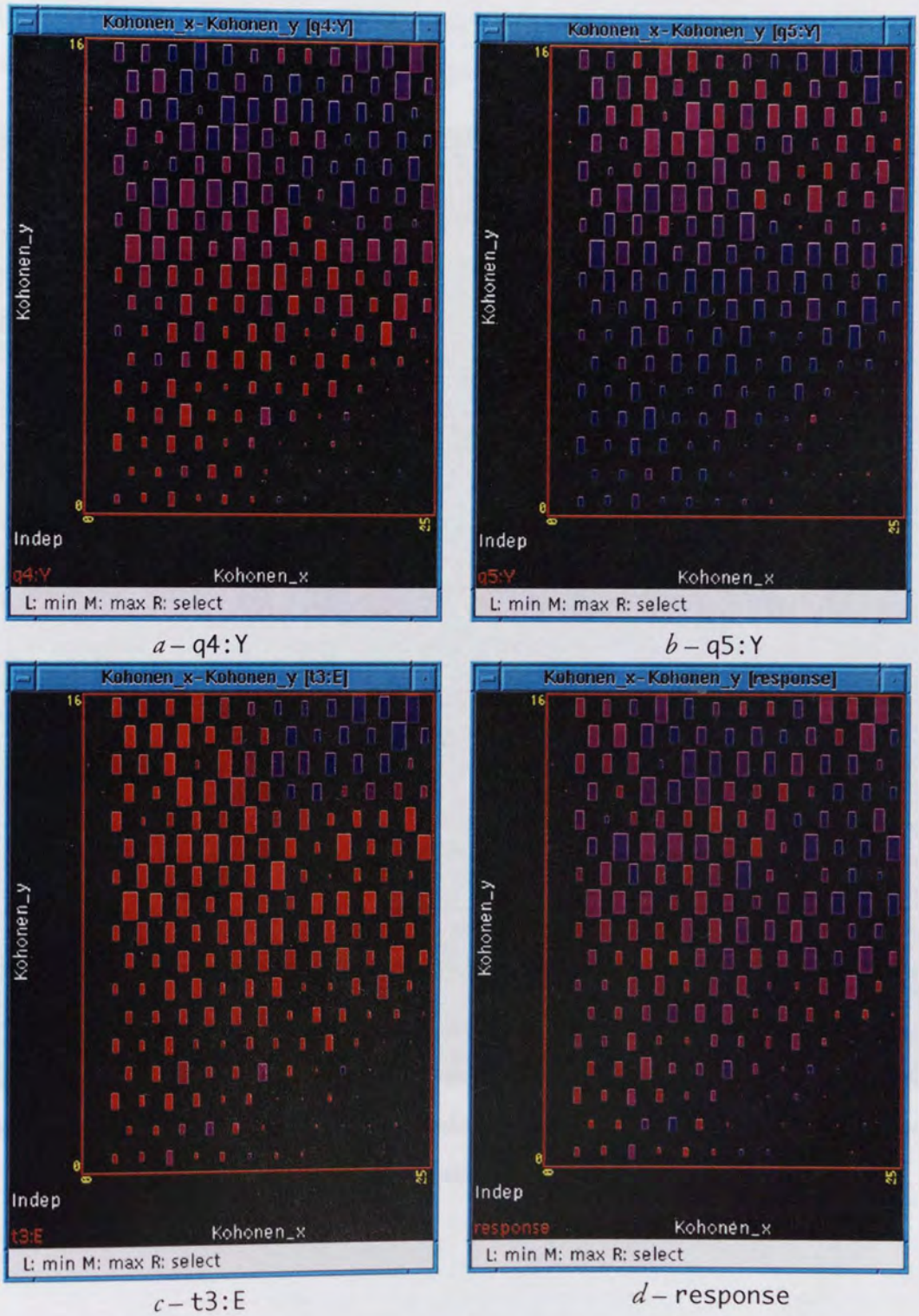
Plate 7.9 – Three-dimensional autoencoding of the mail database, with `Dcard` overlaid

Finally, plate 7.9 shows the area of the three-space where customers with debit cards are positioned. Roughly half of the space is coloured red, with an interesting distribution on the `hidden_1`–`hidden_3` plot.

7.6.2 Finance database

7.6.2.1 Kohonen mapping

Plate 7.10 below shows the distribution of four of the fields of the finance database across its Kohonen-mapped two-dimensional representation. Patterns are evident in the first three, but again there is no clear area of nodes which are active for responders.



a – q4:Y *b* – q5:Y

c – t3:E *d* – response

Plate 7.10 – Kohonen mapping of the finance database with overlays

7.6.2.2 Hidden layer

Plate 7.11 shows the result of training a three-node bottleneck MLP on the finance database. A large amount of internal structure is visible, particularly in the `hidden_1`–`hidden_3` and `hidden_2`–`hidden_3` plots.

The overlay shows the predicted response, which should be compared with plate 7.12 overleaf in which the actual `response` field is overlaid. There is a similarity, but the distinction between red and blue is far less pronounced in the `response` overlay, which shows large areas of well-mixed responders and non-responders.

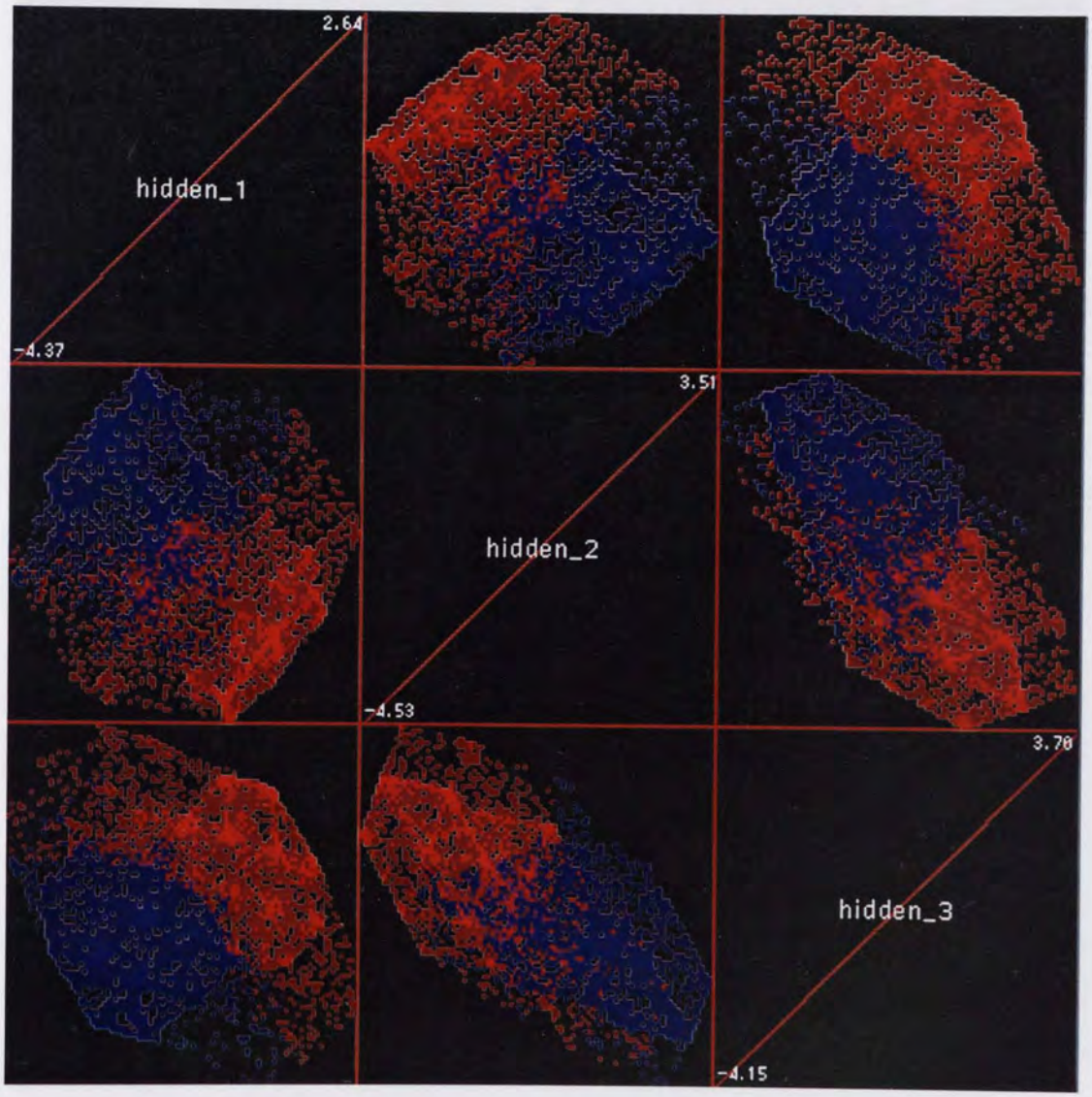


Plate 7.11 – Results of training a three-node bottleneck MLP on the finance database, with predicted response overlaid

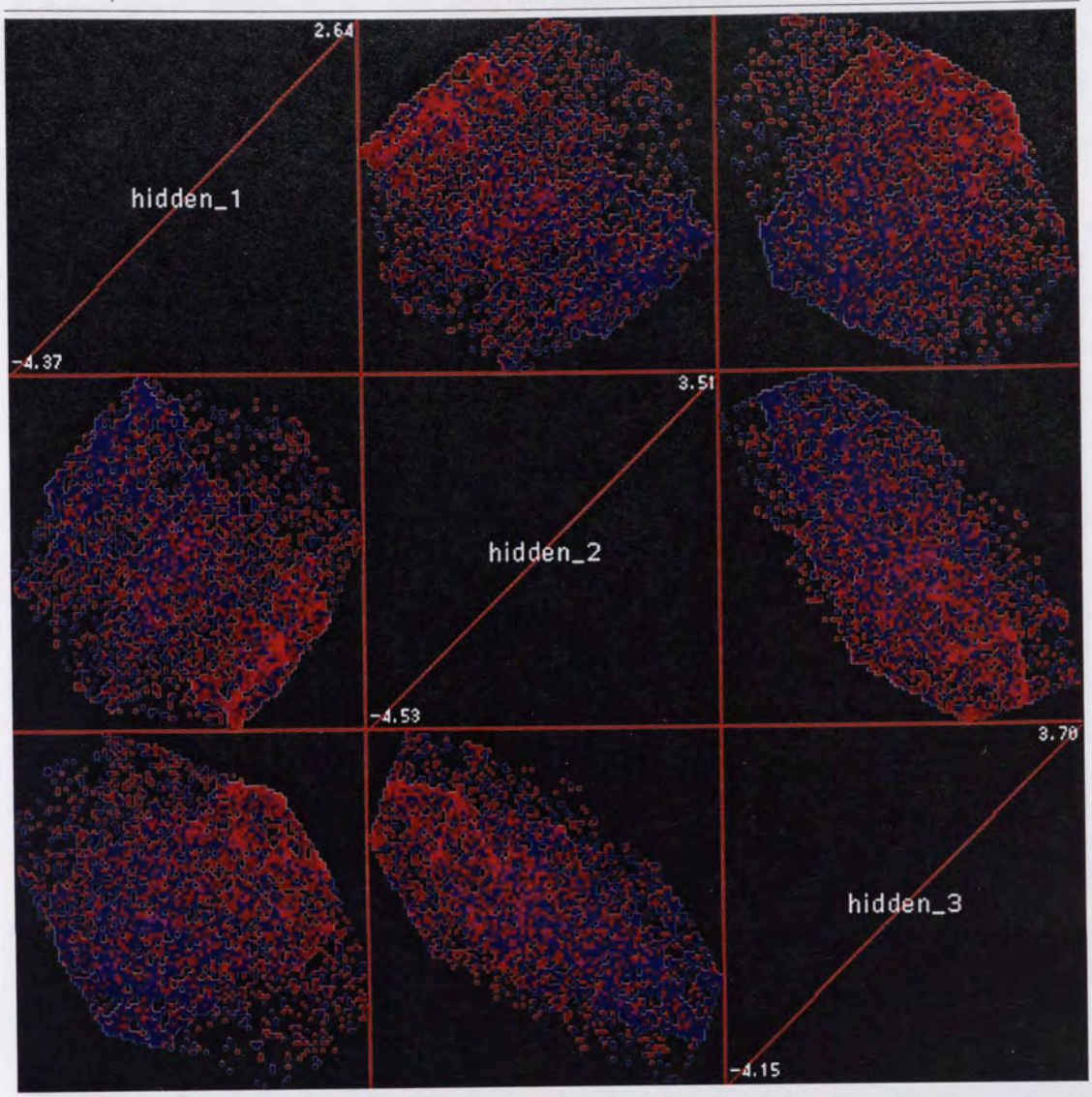


Plate 7.12 – Results of training a three-node bottleneck MLP on the finance database, with actual response overlaid

7.6.2.3 Autoencoder

A three-node autoencoder representation of the finance database in shown below in figure 7.5. Again, considerable structure is evident, particularly in the plot of `hidden_1` against `hidden_2`.

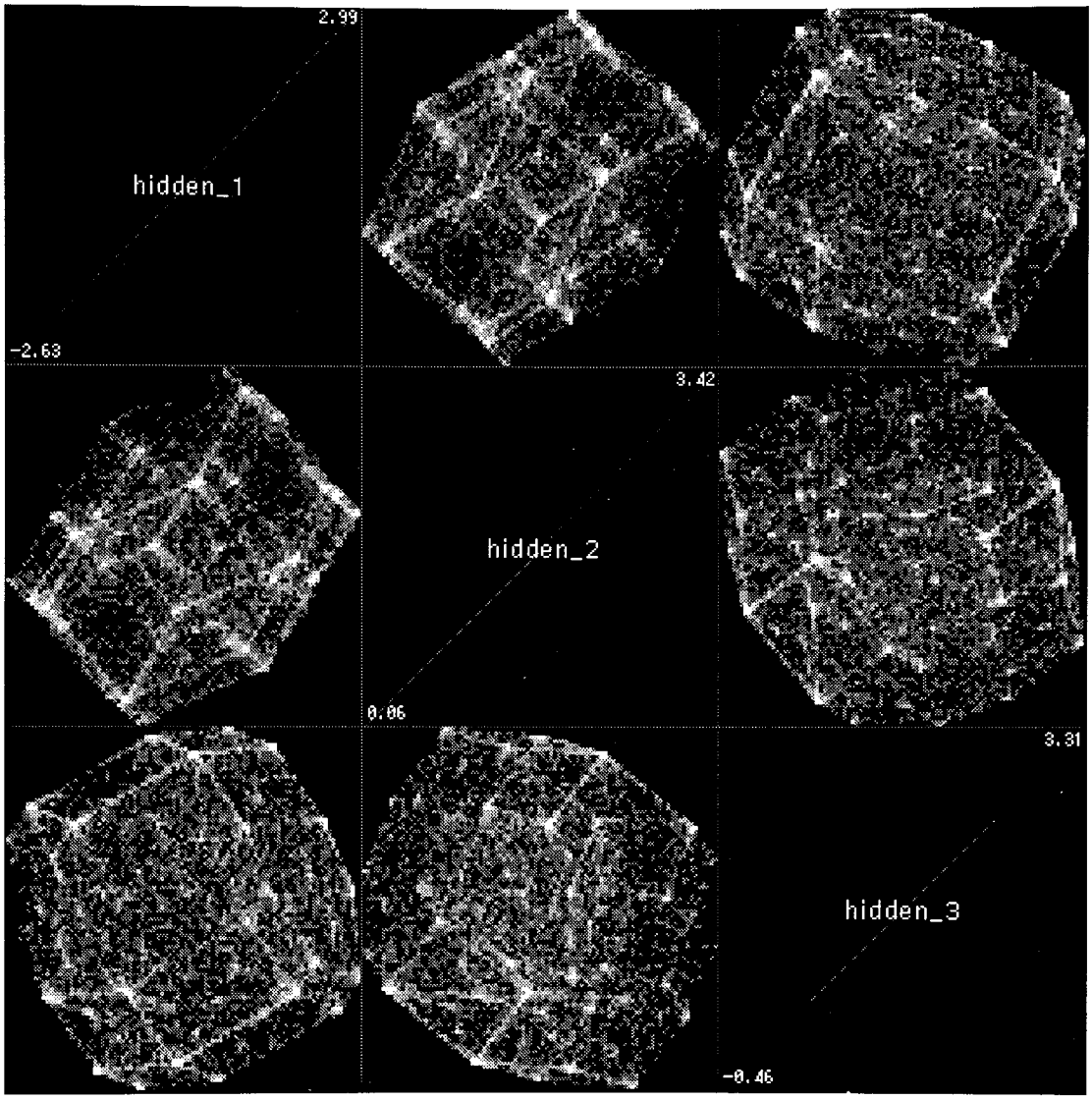


Figure 7.5 – Three-dimensional autoencoding of the finance database

Overlaying the fields of the finance database onto the autoencoding showed similar patterns as seen in the mail database. If the field assignments were known, these results could be interpreted and might yield useful information.

7.6.3 RAE database

7.6.3.1 Kohonen mapping

Figures 7.6 and 7.7 on the following two pages show a series of dependent enlargements following dimensionality reduction of the RAE database using a Kohonen map. The plots are displayed in the same form as the canonical variate plots in the preceding chapter: dependent upon a unit selection on the `#uofa` or `#inst` fields.

Examination of the plots in figure 7.6 yields some interesting results:

- Some disciplines, for example French, Economics, Education and Business & Management, occupy a fairly well-defined area of the map and show considerable internal structure. Others, particularly Biological Sciences, cover a large proportion of the map, with little visible clustering.

- Hospital Clinical, as expected, is unusual; the departments are mapped to four well-separated clusters, three of which are in the corners of the map.

- Most plots show some outlying departments, which can of course be identified using the highlighting mechanism of MADEN. Often, as has been seen in other types of plot, these departments are from Oxford or Cambridge.

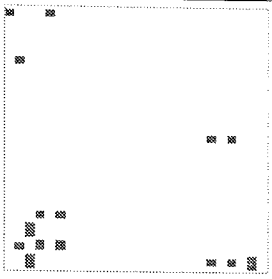
- There is some similarity between Chemistry and Physics, although to a lesser extent than seen with the CV plots.

The institutional plots in figure 7.7 are less revealing, though a few remarks can be made:

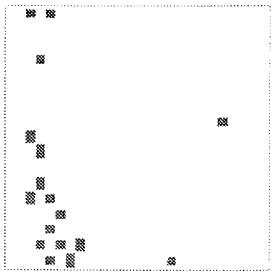
- A few institutions are confined to clear areas of the map, notably Hertfordshire and Ulster, but most are spread across the entire map, showing little obvious structure.

- The 'Oxbridge factor' is less pronounced, though their plots are similar, tending to occupy nodes towards the extremities of the map, avoiding the central nodes.

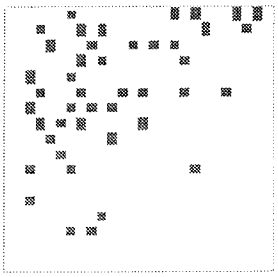
- There is a surprising similarity between the Open University, Stirling and Wales at Aberystwyth.



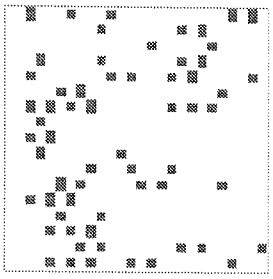
3: Hospital Clinical



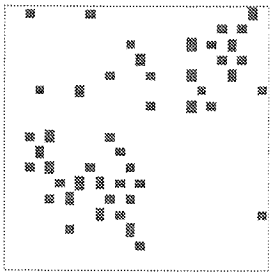
12: Biochemistry



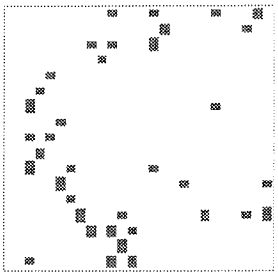
13: Psychology



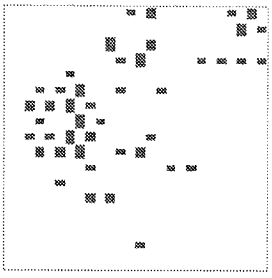
14: Biological Sciences



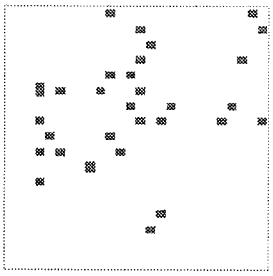
20: Chemistry



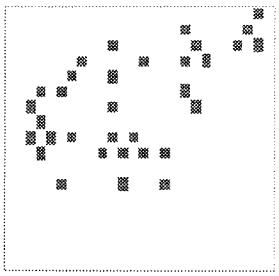
21: Physics



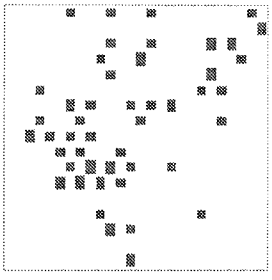
27: Computer Science



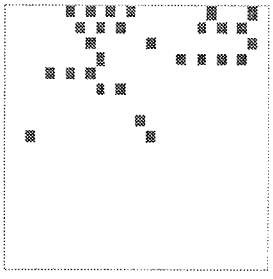
28: General Engineering



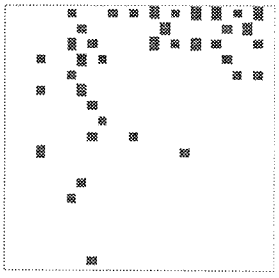
30: Civil Engineering



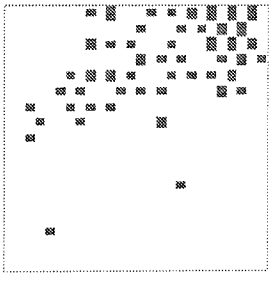
31: Elec Eng



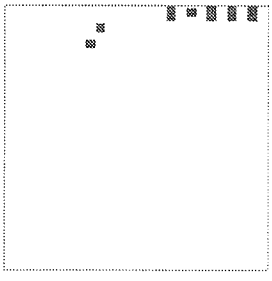
36: Town & Country Planning



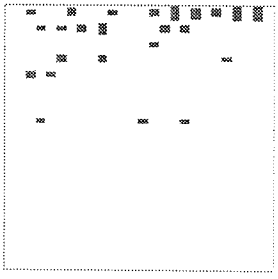
41: Economics



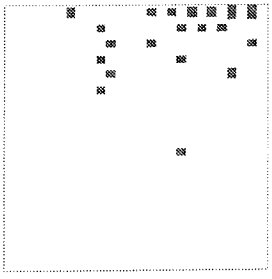
46: Business & Management



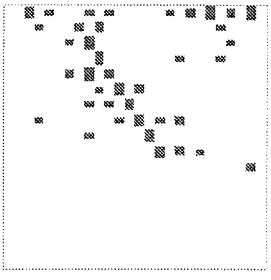
54: French



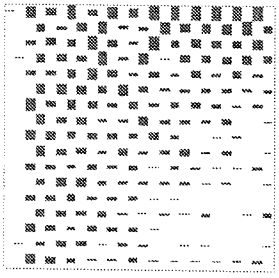
62: History



67: Art & Design



71: Education



All units

Figure 7.6 – RAE Kohonen mapping, dependent upon unit of assessment

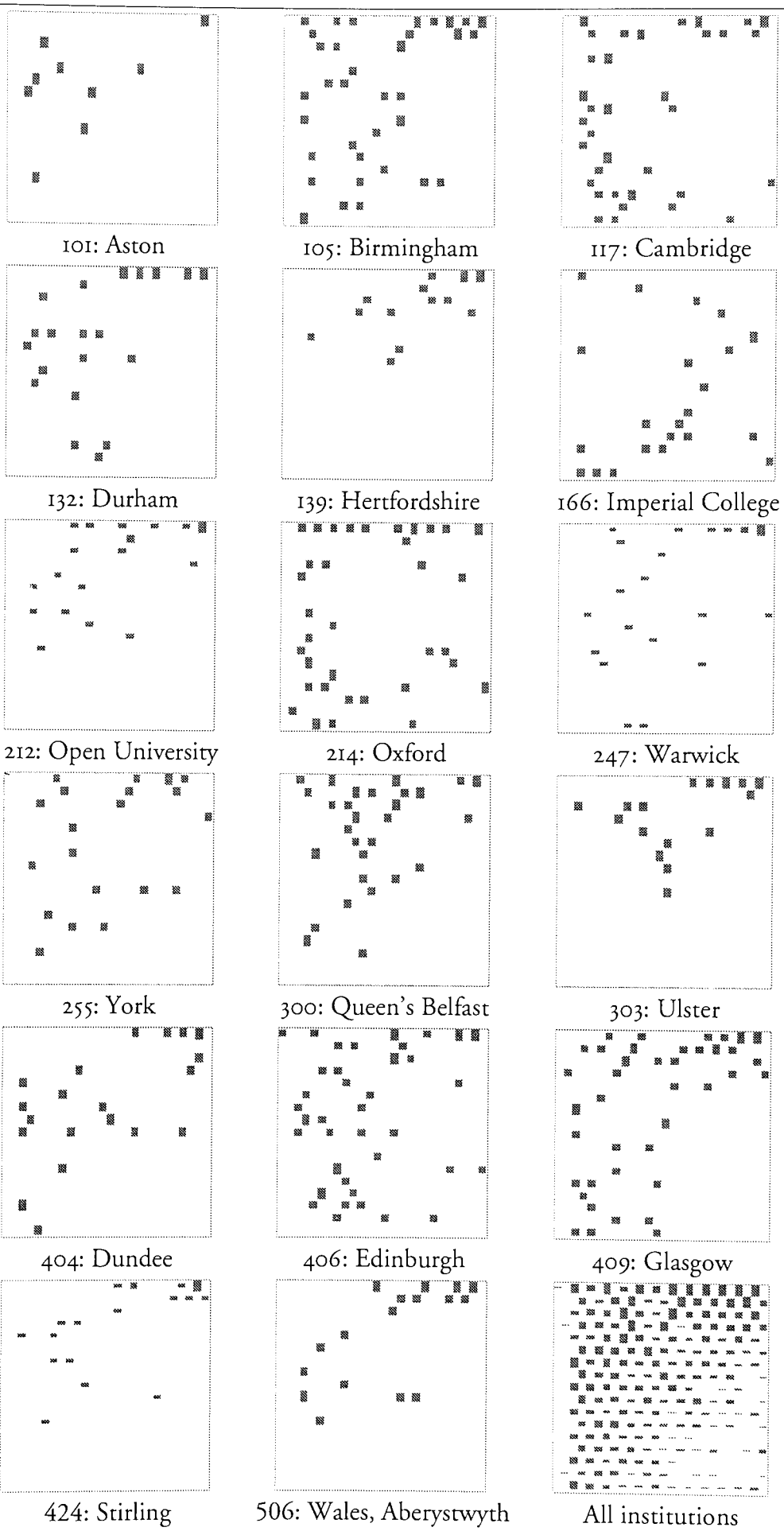


Figure 7.7 -- RAE Kohonen mapping, dependent upon institution

In an attempt to investigate the effect of removing the 'size' component of the RAE data, a Kohonen map was trained on the database standardised by the 'size' factor, as used previously.

Plates 7.13 and 7.14 on the following pages show a series of dependent enlargements as used before, but with the `Rating` field overlaid. Thus the colour of each node indicates the average rating of departments mapped to that node under the Kohonen mapping.

The independent plot (at the bottom right of the two plates) shows a remarkable pattern, with the average rating being very low at two nodes on the left edge, and increasing radially through a wide area of green (average rating three) to highly-rated nodes at the periphery. Clearly there is a definite structure in the standardised data which gives rise to a radial pattern of ratings.

Some conclusions can be drawn from plate 7.13:

- Most disciplines have a more well-defined 'fingerprint' on the map than on the map trained on the unstandardised database, with generally fewer outliers.
- Hospital Clinical and Biochemistry are very different from the other disciplines in two ways: they have no departments mapped to the middle left nodes of the map, and they are split into two clusters.
- Physics seems to be the most unusual discipline shown, with two departments placed at the far right of the map. Unlike in previous plots, it is also quite dissimilar to Chemistry. History and in particular French are once again very nebulous.
- Patterns of ratings can be seen within many plots, for example Civil Engineering and Psychology, but the patterns differ between disciplines.

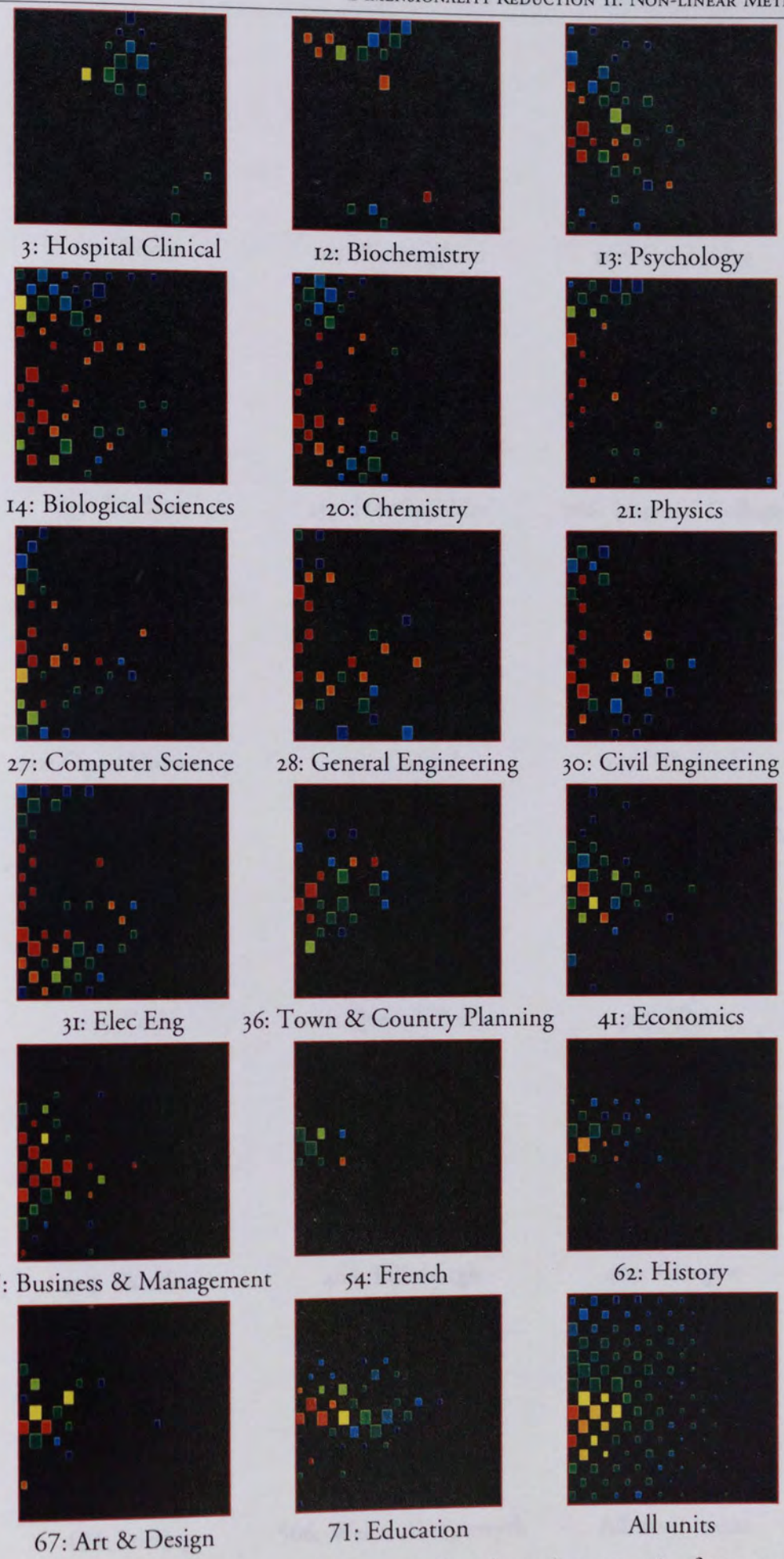


Plate 7.13 – Standardised RAE Kohonen mapping, dependent on unit of assessment

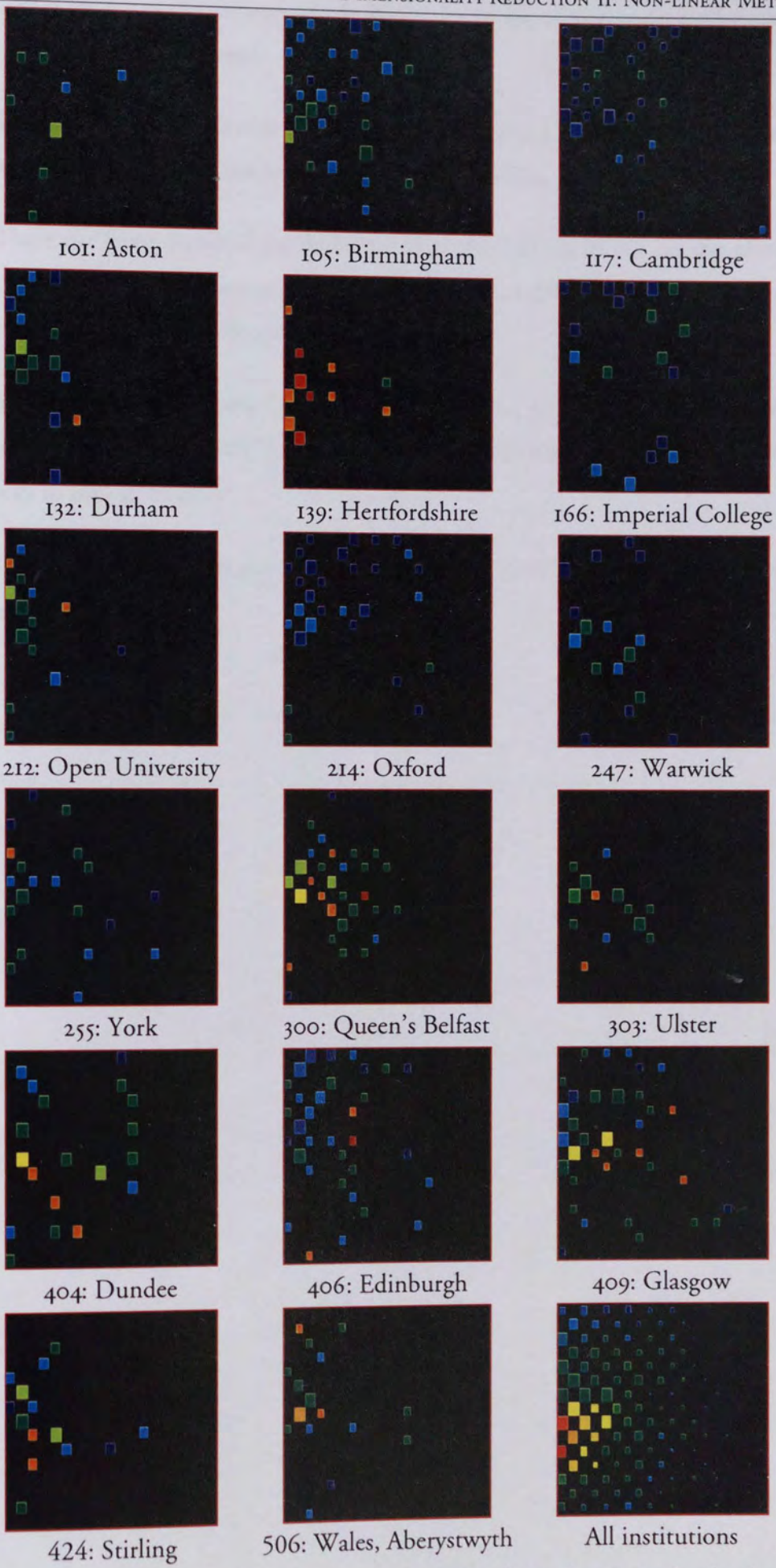


Plate 7.14 – Standardised RAE Kohonen mapping, dependent upon institution

Looking at plate 7.14 allows more conclusions to be drawn concerning the relationships between different institutions:

- The prevalence of dark blue (rated five) nodes in the Cambridge and Oxford plots is striking, as are the low ratings at Hertfordshire.
- There are fewer patterns visible in the placement of the nodes on the plots than in the disciplinary plots; only the Northern Irish plots show a very self-contained region of response.
- Some institutions, notably Durham, Warwick, Aston and the Open University, occupy only the left half of the map, whereas Glasgow reaches nearly all the way to the right side.
- Unlike the disciplinary plots, there is no clear pattern of ratings within most of the plots.

7.6.3.2 Hidden layer

A three-node bottleneck MLP was trained on the RAE database, to predict the rating given the other fields as input. Plate 7.15 below shows the 3-D database constructed from the hidden layer activations, along with the `predicted` field, which is also overlaid.

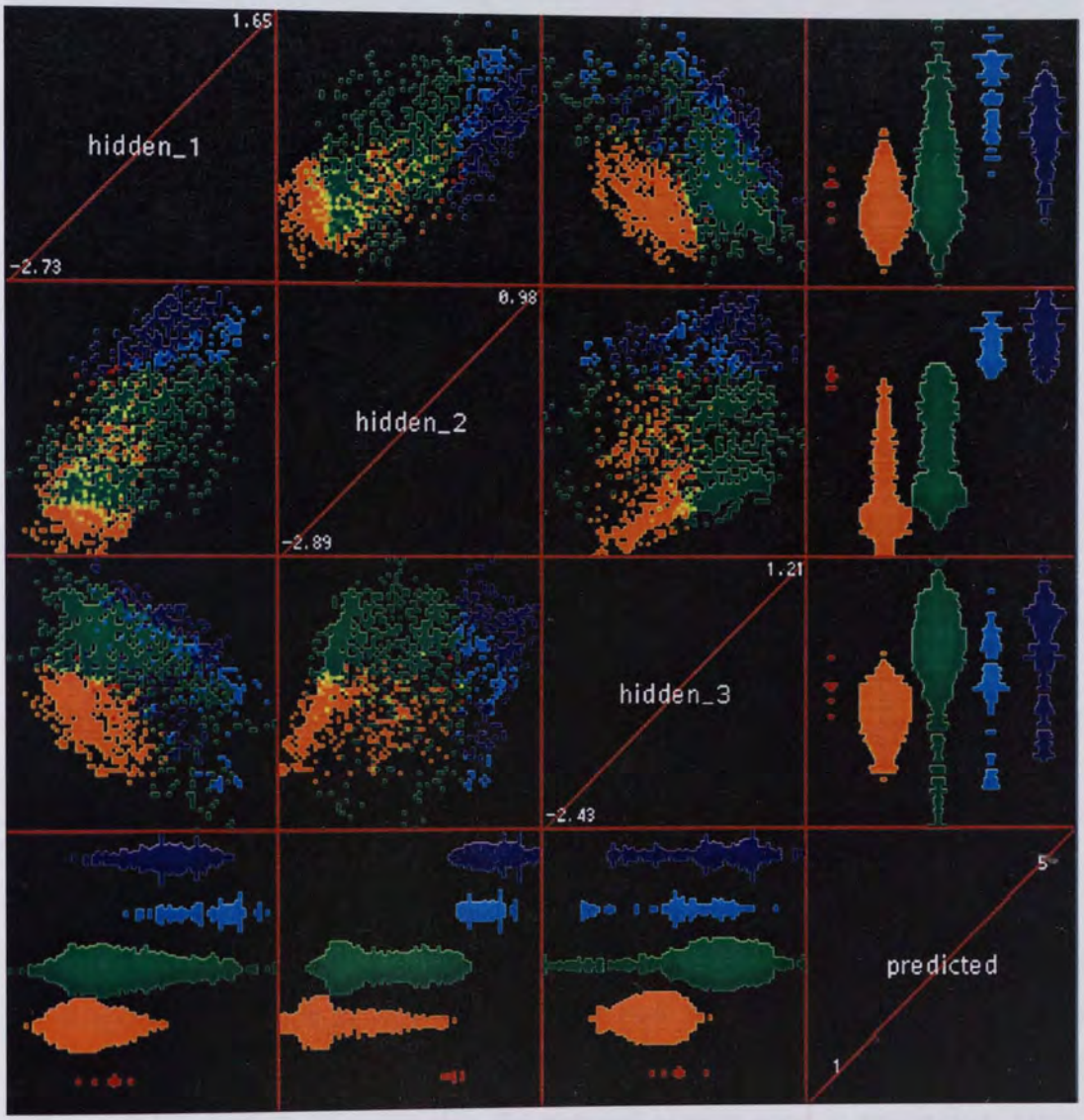


Plate 7.15 – Results of training a three-node bottleneck MLP on the RAE database, with predicted rating overlaid

Clearly, a complex function has been learnt, dividing the 3-D space into five regions corresponding to the five ratings. Notably, very few departments are predicted to get a one rating. This may be due to the limited modelling power offered by three dimensions, or alternatively because the ones are well-mixed and the overall error was lower if they were predicted as twos.

Plate 7.16 overleaf shows the same overview, but with `Rating` overlaid.

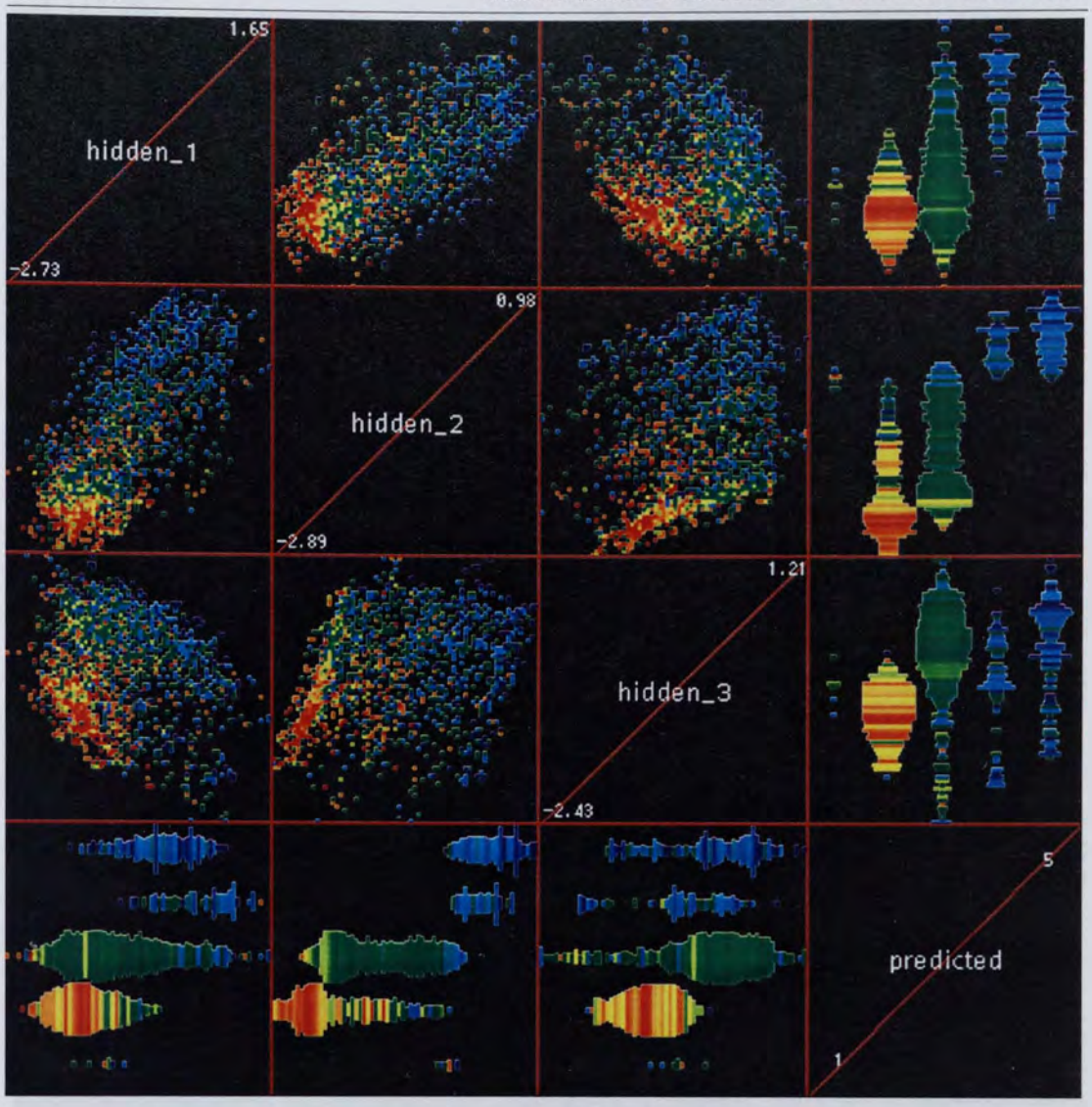


Plate 7.16 – Results of training a three-node bottleneck MLP on the RAE database, with actual `Rating` overlaid

The similarity between plates 7.15 and 7.16 is clear, but there is clearly a large number of misrated departments. Examination of the bottom row of plate 7.16 shows how the departments predicted to get each rating (the rows of the plot) actually fared (the colour of the overlay). It seems that the network is very good at predicting threes, that there is a lot of confusion between fours and fives, and that all the ones were misclassified as twos.

Evidently MADEN can be successfully applied to the visual analysis of the performance of neural networks

6.3.3 Autoencoder

Finally, the output of a three-dimensional autoencoder is shown in figure 7.8. This plot shows little of interest, particularly in comparison with the autoencodings of the mail and finance databases. The results of overlaying fields of the database were not particularly revealing, and are not shown here.

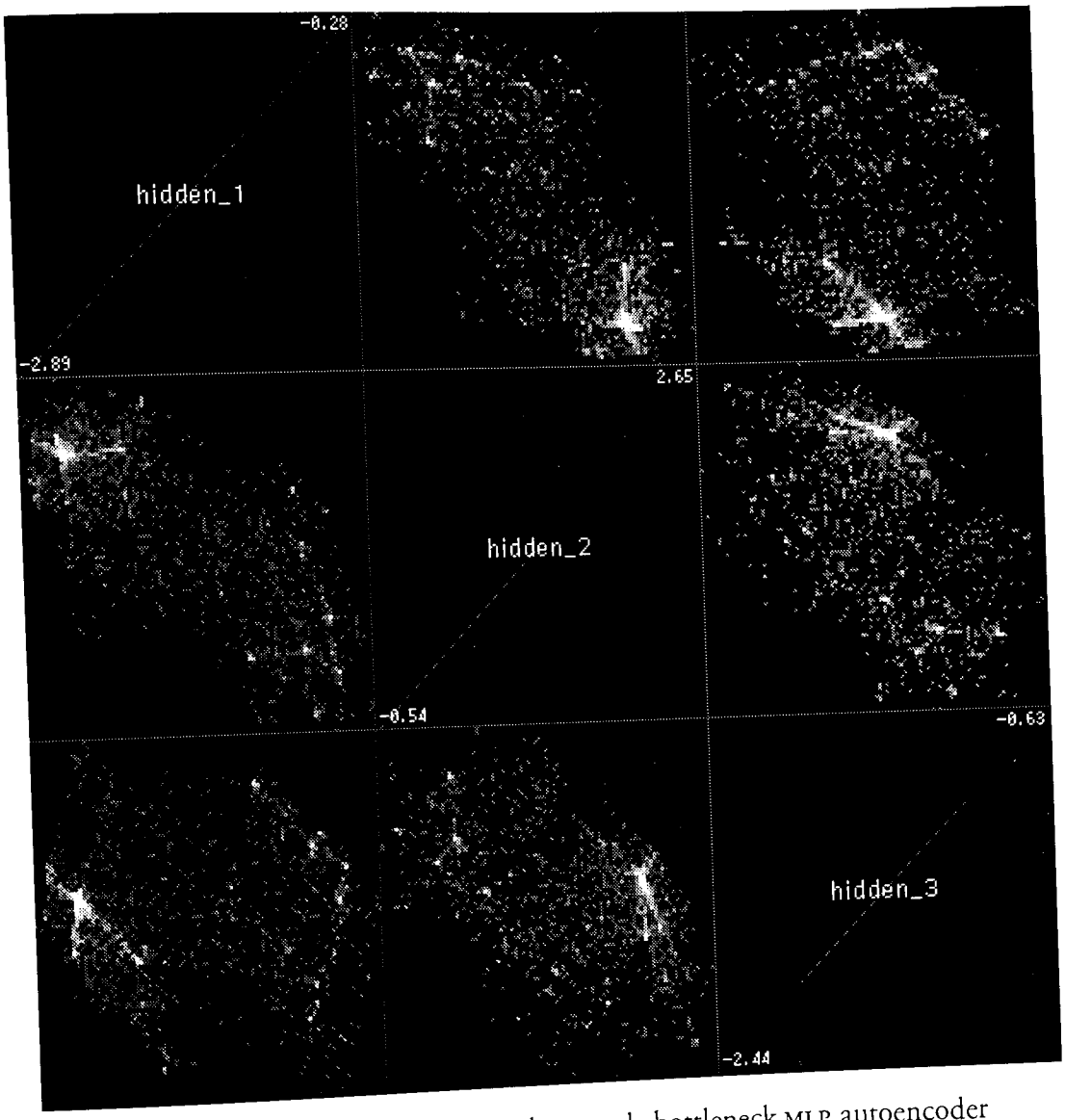


Figure 7.8 – Results of training a three-node bottleneck MLP autoencoder on the RAE database

7.7 Conclusions

Non-linear dimensionality reduction techniques, while appearing to offer a range of possibilities for data analysis in MADEN, were found to give mixed performance.

The use of the Kohonen self-organising map for dimensionality reduction transforms the entire database onto a 2-D lattice of a few hundred nodes. In itself, this does not reveal a lot of information about the data, but by overlaying fields of the original database to look for patterns across the map, and by using dependent enlargements to examine where particular data records are placed, the mapping provides useful tools for analysing the data in two dimensions.

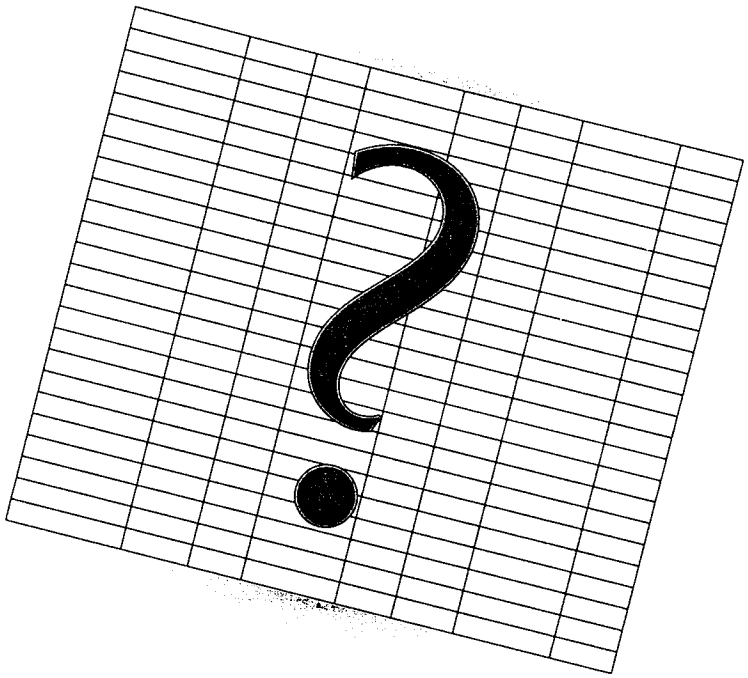
The 'bottleneck' multi-layer perceptron network which learns to predict the response variable of a database is attractive in principle, and in practise often generates interesting views of the data in two or three dimensions, in which the variation of response can clearly be seen. However, the use of the MLP in MADEN is maybe of more use in examining how the neural network behaves than in exploring a particular database.

The autoencoder network has been demonstrated to have the ability to reveal remarkable, almost crystalline, structures within databases, which would probably appear even more interesting in a true 3-D viewing system. In conjunction with the use of overlays it allows the user to investigate how the network combines fields to generate an encoded representation of the data, but, again, it does not add a lot of data analysing power to MADEN.

The major drawback of the three non-linear techniques which were implemented is the time each takes to generate a low-dimensional representation of the database. In particular, it can take several hours to train an MLP on a large database, running on a moderately powerful workstation. Coupled with the generally poor performance of the MLP methods, it seems that only the Kohonen map has anything genuinely useful to offer to a user of MADEN.

Several promising non-linear techniques available were found to be inapplicable due the size of the databases under consideration here. Given faster machines with greater memory capacity, it would be interesting to apply such methods as NLM, NEUROSCALE and MDS to the example databases.

Chapter 8



Conclusions

What we call the beginning is often the end
And to make an end is to make a beginning.
The end is where we start from.

[Eliot, 1942]

8.1 Conclusions

8.1.1 The Benediktine cyberspace cell

The Benediktine cell, the initial idea which sparked this research, was (and is) an attractive concept. The representation of real data as objects (in this case decorated walls) in a completely virtual cyberspace encapsulates visualisation. The implementation of the cell and its subsequent development proved that Benedikt's proposition was to a large extent valid, and could indeed form the basis of a useful visualisation system.

The original purpose of Benedikt's cell was to navigate through a library of images, using subspaces to open the next level of the search, resulting finally in the actual images on the walls of the room. The implementation described here uses the same type of cell for exploring a different sort of database, which helps to explain some of the difficulties with the system.

8.1.2 Maden

MADEN, the novel visualisation system based on two-dimensional matrices of density plots, in general succeeded in overcoming most of the limitations of the cell implementation, and has been developed into a powerful tool for investigating databases. It allows an entire database to be visualised in one window, and individual 2-D plots to be enlarged into separate windows for close scrutiny. Multidimensional selections can be made and examined, and individual data records can be highlighted and identified. More importantly, the architecture is such that a number of data processing techniques could easily be integrated into MADEN in a consistent and accessible form.

8.1.3 Data reduction

Data reduction at first appeared to be vital for visualising databases of the size under consideration. However, the nature of the databases hindered most of the data clustering techniques which were attempted. The Kohonen self-organising map was the only consistently useful technique. Not only does it dramatically reduce the size of the database, but also, through examination of the weight vector components across the map, allows observations to be made about the structure and shape of the data.

8.1.4 Linear dimensionality reduction

Linear dimensionality reduction – finding a small number of directions through the data space and projecting the database onto them – provided a wide range of tools for data analysis. In testing, principal factor analysis generated the same variance-maximising axes as (undirected) principal component analysis, though taking a considerably longer time to do so. The 'directed PCA' technique was proposed to allow a response variable in a database to influence the choice of axis, and had some success, particularly with the otherwise difficult finance database.

The use of projection pursuit methods to automatically determine an 'interesting' pair of projection axes is an attractive concept, but the implementation gave a disappointingly poor performance on the large databases under consideration. Linear discriminant analysis, however, proved to be an excellent tool for analysis of data which is split into groups. Results from the application of LDA to the RAE database were startling, as will be discussed below.

8.1.5 Non-linear dimensionality reduction

Applying the few non-linear dimensionality reduction techniques which were feasible for large databases gave mixed results. Non-linear discriminant analysis and non-linear principal component analysis both tended to illuminate the neural networks in use rather than the databases themselves, but once again, the Kohonen map proved to be useful. It allows the data to be mapped to a two-dimensional grid, with the possibility of overlaying data fields such as a response variable, and in conjunction with the use of dependent enlargements, is an extremely powerful tool.

8.1.6 Summary

The aim of this research was set out in section 1.1.2:

> *...[to develop] advanced tools to allow a database to be displayed on a computer screen in such a way that the user of the software can gain insights into the contnets of the datbase through interactivly 'exploring' it in some fashion.*

I believe this aim has been achieved. The MADEN system allows a non-specialist (with a small amount of training) to see, understand and explore a tabular database, whether in its entirety or in close detail. The numerous data processing options – many of which, including the novel DPCA technique and the various neural networks, have been applied to database visualisation for the first time – permit the data to be visualised from radically different viewpoints, often highlighting particular features of the data which were previously hidden.

The following section summarises the results obtained from the three example databases, and demonstrates that different databases often respond with more useful information when processed in different ways. MADEN offers a suitably diverse set of tools to the user, each useful in certain situations.

8.2 Summary of Results

8.2.1 Mail database

Visualisation of the 'raw' mail database using both the Benediktine cell and MADEN helped to understand its structure (e.g. the meaning of the categories of the 'home' field) and to see some interesting relationships between pairs of fields (e.g. the two clusters on the plot of minimum balance against account turnover). Overlaying the response field indicated where the areas of high response lay, leading to the discovery that the 'has life insurance' field segregates a great number of non-responders, and using dependent enlargements showed how certain patterns changed with customer age. By making a selection along three axes, it was possible to increase the response rate from 50% to 86%, albeit at the cost of eliminating many responders.

Clustering of the database using the Kohonen self-organising map allowed segments of the customers to be discerned, by combining several weight vector components. Principal component and factor analysis gave a set of axes through the data space, most of which had a clear interpretation, for example a 'wealth index'. Projections onto these axes revealed yet more clustering in the database, though most of this was probably due to the discrete nature of the fields involved.

Projection pursuit using the skew index resulted in a very interesting plot, with numerous clusters, some of which contained very large or very small numbers of responders. Linear discriminant analysis achieved a 31% separation between responders and non-responders, mainly through the life insurance information.

Non-linear dimensionality reduction revealed a little more about the database. The Kohonen mapping showed more patterns in the data, though its main result was to show that the data itself was far less clearly grouped than the weight vector components. Non-linear discriminant analysis showed how a bottleneck neural network could learn to predict customer response. This process, and the autoencoder network, also revealed unexpectedly intricate internal structure in the database. By overlaying database fields onto the autoencoder representation, it was possible to see where certain customer groups (for example those who rent their homes) were placed in the three-dimensional encoded space.

8.2.2 Finance database

It was difficult to draw conclusions from the results of analysing the finance database, simply due to the lack of information regarding field assignments.

Numerous correlations and 'interesting' plots have been presented, and some very detailed internal structure was brought out by the use of the two neural networks. However, the most interesting plot was generated by the novel directed principal component analysis technique, which was able to separate responders from non-responders in a much more powerful way than any other method.

8.2.3 RAE database

Visualisation of the raw RAE database was difficult, due to the large number of fields and the wide ranges of values on most of them. The addition of the alternate colour scale allowed the five ratings to be visually separated, and the development of the highlighting operation permitted individual departments to be identified by name.

Clustering with the Kohonen map allowed some patterns in the data to be seen, for example the departments with high numbers of 'cited publications classed as applied research' were grouped into four areas of the map – although some well-separated map nodes had to be clipped out before such patterns became observable.

The size of the RAE database made dimensionality reduction almost vital for analysis, and the results from dimensionality-reducing methods were quite remarkable. Factor analysis revealed that the most significant factor is closely related to the size of the department. Standardising the database by dividing by this size measure allowed conclusions to be drawn regarding the relationship between the relative size of certain fields and the resulting rating. For example, having more publications for a given 'size' tends to increase the department's rating.

Projection pursuit could only work with a small subset of the database fields, and when applied to the 'people-related' fields of the database standardised by the 'size' factor, resulted in a plot with a clear pattern of ratings – even though the projection-finding algorithm did not have any knowledge of the ratings awarded.

Linear discriminant analysis revealed that there is a near-linear continuum along which ratings range from one to five. Departments which rate a one or a five are

generally quite separated from the middle rating departments, which are more tightly clustered. Using dependent enlargements, the canonical variate plots for individual institutions and disciplines were generated and compared. Observations included the fact that Hospital Clinical was particularly different from the other disciplines – indeed, this is one of the disciplines which are being separately analysed in the next assessment exercise – and that Physics and Chemistry are quite similar.

Use of the Kohonen map for dimensionality reduction provided another set of plots to compare disciplines and institutions, together with their average ratings. In this case, many patterns of map locations and ratings could be seen. French departments were seen to be very similar to one another, but in this case Physics was quite different from Chemistry. Unexpected similarities between institutions (Open, Stirling and Wales at Aberystwyth) were seen, and an unusual institution (Glasgow) identified. Non-linear discriminant analysis showed how the three-dimensional space of the bottleneck nodes of the neural network was partitioned into five ratings, and demonstrated where the network was correct, or, more often, incorrect in its prediction.

There may be said to be two major results from this analysis of the RAE92 database:

Firstly, the data supplied for the research assessment exercise can be used to predict the research rating with some degree of accuracy. In particular, simply by calculating and dividing by the 'size' factor, and projecting onto the two linear discriminant axes given in chapter 6, one can say with a degree of confidence whether the department in question will rate a one, or a four/five. As might be expected, the middle ratings are more difficult to predict.

Secondly, the 'Oxbridge factor' is not purely an imaginary distinction – Oxford and Cambridge have been shown to be clearly very different from other institutions based on purely empirical data. Of course, the data itself may have arisen due to the reputation of the universities bringing large numbers of grants etc.

8.3 Further Work

Inevitably, at the end of a project such as this, there is a long list of 'wibni's ('wouldn't it be nice if...'). Some are major avenues which were not explored, while some are (seemingly) small modifications to the user interface which were either suggested by other users, or which have begun to annoy – but not quite enough to devote any time to fixing them. In no particular order:

- Further research should be carried out into the automation of exploration. As has been noted, different databases require different tools. It should be possible to let the system 'guess' which tools will be of most use – maybe initially by trying them all and presenting the collected results to the user.

- Certain processing operations could be augmented to display their progress graphically. This would add a lot to the system in terms of interaction: instead of simply pressing a button and waiting for a result to appear, the user could actually watch clusters forming, neural networks training, projection pursuit in action etc. Obviously there would be a trade-off between the amount of feedback given and the speed of operation, but a good compromise could be reached.

- Axes such as principal components might be made easier to 'read' by ordering their components by the absolute size of each component. Also, a method of generating a textual summary of each discovered linear axis in terms of its largest components could be developed, and used to prompt the user for an interpretive name, rather than the uninformative 'PC_1' etc.

- Serial number fields need to be expanded to handle textual fields (e.g. customer names). Highlighting records could then display textual information.

- When exploring the mail database, it might be useful to be able to 'normalise' the density plots against Age, in order to remove the effect of the 'bulge' in the age density at around 50 years. Exactly how this would be implemented is unclear: research is required.

List of References

Adams, Douglas. *The Hitch-Hiker's Guide To The Galaxy.* 1980.

Aldenderfer, Mark S & Roger K Blashfield. *Cluster Analysis.* Beverley Hills: SAGE Publications ISBN 0803923767, 1984.

Allen, R E, ed. *The Concise Oxford Dictionary of Current English.* 8th ed., Oxford: Clarendon Press ISBN 0198612001, 1990.

Andrews, D F. "Plots of high dimensional data." *Biometrics* 28 (1972): 125–36.

Ballé, Michael & Trevor Jones. "Data Visualisation: A Fresh Look at Segmentation." *Journal of Database Marketing* 1, no. 1 (1993): 62–76.

Becker, Richard A & William S Cleveland. "Brushing Scatterplots." *Technometrics* 29 (1987): 127–142.

Becker, Richard A, William S Cleveland & Allan R Wilks. "Dynamic Graphics for Data Analysis." In *Dynamic Graphics for Statistics,* ed. William S Cleveland & Marylyn E McGill. New York: Wadsworth, 1988.

Benedikt, Michael, ed. *Cyberspace: First Steps.* Cambridge, Mass.: MIT Press ISBN 02620237X, 1991A.

Benedikt, Michael. "Cyberspace: Some Proposals." In *Cyberspace: First Steps,* ed. Michael Benedikt. Cambridge, Mass.: MIT Press ISBN 02620237X, 1991B.

Benford, S, J Boyle, R Cooper, P Gray *et al. Experience of Using 3-D graphics in Database Visualisation.* 1994.

Berger, Marc. *Computer Graphics with Pascal.* Menlo Park, CA: Benjamin/Cummings ISBN 0803307907, 1986.

Berkeley, George. *A Treatise Concerning the Principles of Human Knowledge.* Introduction, sect. 3, 1710.

Bertin, Jacques. *Graphics and Graphic Information-Processing.* Translated by William J Berg & Paul Scott. Berlin: Walter de Gruyter ISBN 3110088681, 1981.

Bier, Eric A, Maureen C Stone, Ken Fishkin, William Buxton & Thomas Baudel. "A Taxonomy of See-Through Tools." In *Proceedings of CHI '94* (New York). ACM, 1994.

Bounds, David & Philip Barrett. "Neural Networks and Data Visualisation." In *Neural Networks,* ed. J G Taylor. Henley on Thames: Alfred Waller, 1995: 149–164.

Boyle, John. "Information Visualisation Systems that use 3-D Graphics." *ACM Transactions on Information Systems* (to appear 1995).

Card, Stuart K, George G Robertson & Jock D Mackinlay. "The Information Visualizer, An Information Workspace." In *Proceedings of CHI '91*. ACM, 1991: 181–188.

Carr, Daniel B. "Looking at large data sets using binned data plots." In *Computing and Graphics in Statistics*, ed. Andreas Buja & Paul A Tukey. New York: Springer-Verlag ISBN 0387976337, 1991.

Chambers, J M, W S Cleveland, B Kleiner & P A Tukey. *Graphical Methods for Data Analysis.* Belmont, CA: Wadsworth ISBN 053498052X, 1983.

Chatterjee, Avijit. "Parallel Visual Explorer at work in the Money Markets." IBM press release, Feb 2nd 1995.

Chernoff, H. "The Use of Faces to Represent Points in *k*-Dimensional Space Graphically." *Journal of the American Statistical Association* 68 (1973): 361–368.

Clarkson, Mark A. "An Easier Interface." *Byte,* February 1991: 277–282.

Cleveland, William S & Robert McGill. "Graphical perception, theory, experimentation and application to the development of graphical methods." *Journal of the American Statistical Association* 79, no. 387 (1984): 531–554.

Cochrane, Peter. "The Virtual University." Presentation given at Aston University, March 1995.

Conner, D Brookshire, Scott S Snibbe, Kenneth P Herndon, Daniel C Robbins *et al.* "Three-dimensional Widgets." In *1992 Symposium on Interactive 3-D Graphics* (Cambridge, MA). ACM, 1992: 183–188.

Crichton, Michael. *Disclosure.* London: Random House ISBN 067419454, 1993.

Csinger, Andrew. *The Psychology of Visualization.* Dept of Computer Science, University of British Columbia, 1992.

Davison, Mark L. *Multidimensional Scaling.* New York: John Wiley & Sons ISBN 047186417X, 1983.

DeMers, David & Garrison Cottrell. "Non-linear Dimensionality Reduction." In *Advances in Neural Information Processing Systems 5*, ed. C L Giles, S J Hanson & J D Cowan. San Mateo: Morgan Kaufmann, 1993: 580–587.

de Saint-Exupéry, Antoine. *Le Petit Prince.* Chapter 21, 1943.

Edwards, Greg. "Visualisation: the second generation." *Image Processing*, May/June 1992: 48–53.

Eliot, T S. "Little Gidding." In *Four Quartets,* 1942.

Ellis, Stephen R, Mary K Kaiser & Arthur J Grunwald, ed. *Pictorial Communication in Virtual and Real Environments.* London: Taylor & Francis, 1991.

Everitt, Brian S. *Cluster Analysis.* London: Heinemann, 1974.

Everitt, Brian S. "A Finite Mixture Model for the Clustering of Mixed-mode Data." *Statistics & Probability Letters* 6 (1988): 305–309.

Everitt, Brian S & C Merette. "The clustering of mixed-mode data: a comparison of possible approaches." *Journal of Applied Statistics* 17, no. 3 (1990): 283–297.

Everitt, Brian S & Graham Dunn. *Applied Multivariate Data Analysis.* London: Edward Arnold, 1991.

Farquhar, A B & H Farquhar. *Economic and Industrial Delusions: A Discourse of the Case for Protection.* New York: Putnam, 1891.

Feiner, Steven & Clifford Beshers. "Visualizing n-Dimensional Virtual Worlds with n-Vision." *Computer Graphics* 24, no. 2 (1990): 37–38.

Friedman, J H & J W Tukey. "A projection pursuit algorithm for exploratory data analysis." *IEEE Transactions on Computing* C-23 (1974): 881–889.

Furnas, George W & Andreas Buja. "Prosection Views: Dimensional Inference through Sections and Projections." *Journal of Computational and Graphic Studies* 3, no. 4 (1994): 323–353.

Gelernter, David. *Mirror Worlds.* New York: Oxford University Press, 1991.

Gibson, William. *Neuromancer.* New York: Ace Books ISBN 0441569595, 1984.

Globus, Al & Eric Raible. "Fourteen Ways To Say Nothing With Scientific Visualization." *Computer* 27, no. 7 (1994): 86–76.

Hartigan, John A. *Clustering Algorithms.* New York: John Wiley & Sons, 1975.

Hastie, T & W Stuetzle. "Principal Curves." *Journal of the American Statistical Association* 84 (1989): 502–516.

Herman, G T & H Levkowitz. "Color Scales for Image Data." *IEEE Computer Graphics and Applications* 12, no. 1 (1992): 72–80.

Holsheimer, Marcel & Arno Siebes. *Data Mining: The Search for Knowledge in Databases.* CWI, Amsterdam (Ref: CS-R9406), 1994.

Huber, Peter J. "Projection Pursuit." *The Annals of Statistics* 13, no. 2 (1985): 435–475.

Jog, Ninad & Ben Shneiderman. *Interactive Smooth Zooming in a Starfield Information Visualisation.* Human Computer Interaction Laboratory, University of Maryland (Ref: CS-TR-3286), 1994.

Johnes, Jill, Jim Taylor & Brian Francis. "The Research Performance of UK Universities: a Statistical Analysis of the Results of the 1989 Research Selectivity Exercise." *Journal of the Royal Statistical Society* A, no. 156, Part 2 (1993): 271–286.

Jones, M C & Robin Sibson. "What is Projection Pursuit?" *Journal of the Royal Statistical Society A* 150, no. 1 (1987): 1–36.

Kaiser, H F. "The varimax criterion for analytical rotation in factor analysis." *Psychometrika* 23 (1958): 187–200.

Keim, Daniel A & Hans-Peter Kriegel. "VisDB: Database Exploration Using Multidimensional Visualization." *IEEE Computer Graphics and Applications* 14, no. 5 (1994): 40–49.

Kohonen, Teuvo. "The Self-Organising Map." *Proceedings of the IEEE* 78, no. 9 (1990): 1464–1480.

Kohonen, Teuvo, Jussi Jynninen, Jari Kangas & Jorma Laaksonen. *SOM_PAK: The Self-Organising Map Program Package.* Helsinki University of Technology, 1995.

Kramer, Mark A. "Nonlinear Principal Component Analysis Using Autoassociative Neural Networks." *AIchE Journal* 37, no. 2 (1991): 233–243.

Krzanowski, W J. *Principles of Multivariate Analysis: A User's Perspective.* Oxford: Clarendon Press, 1988.

Lee, Melissa. "Virtual reality: toy or tool?" *Chemistry in Britain,* June 1993, 455–456.

Lesser, Michael. *GIFIC.* Personal communications, May 1995.

Levkowitz, Haim & Ronald M Pickett. "Iconographic integrated displays of multiparameter spatial distributions." *SPIE Human Vision and Electronic Imaging: Models, Methods and Applications* 1249 (1990): 345–355.

Little, Roderick J A & Donald B Rubin. *Statistical Analysis with Missing Data.* New York: John Wiley & Sons, 1987.

Lowe, David & Michael Tipping. "A novel neural network technique for exploratory data analysis." In *Proceedings of ICANN '95* (to appear), 1995.

Mackinlay, Jock D, George G Robertson & Stuart K Card. "The Perspective Wall: Detail and Context Smoothly Integrated." In *Proceedings of CHI '91.* ACM, 1991: 173–179.

MacQueen, J. "Some methods of classification and analysis of multivariate observations." In *Proceedings of the 5th Berkeley Symposium on Mathematics, Statistics and Probability* L M LeCam & J Neyman ed. University of California Press, 1967: 281–297.

Malthouse, Edward Carl. "Nonlinear Partial Least Squares." PhD thesis, Northwestern University, Illinois, 1995.

Mao, Jianchang & Anil K Jain. "Artificial Neural Networks for Feature Extraction and Multivariate Data Projection." *IEEE Transactions on Neural Networks* 6, no. 2 (1995): 296–317.

Mardia, K V, J T Kent & J M Bibby. *Multivariate Analysis.* London: Academic Press, 1979.

Mariani, John A & Robert Lougher. "TripleSpace: an experiment in a 3-D graphical interface to a binary relational database." *Interacting with Computers* 4, no. 2 (1992): 147–162.

McCormick, B, T A DeFanti & M D Brown. "Visualization in scientific computing." *Computer Graphics* 21, no. 6 (1987).

McLachlan, Geoffrey J & Kaye E Basford. *Mixture Models: Inference and Applications to Clustering.* New York: Marcel Dekker, 1988.

Mihalisin, T, J Timlin & J Schwegler. "Visualisation and analysis of multi-variate data: a technique for all fields." In *Proceedings of Visualisation '91* (San Diego, CA), IEE, 1991: 171–178.

Milne, A A. "Disobedience." In *When We Were Very Young,* 1924.

Milne, A A. "Introduction." In *Now We Are Six,* 1927.

Neesham, Claire. "Seeing Reason." *Computing,* 29 April 1993, 16–17.

Pratchett, Terry. *Pyramids.* London: Corgi, 1989.

Press, William H, Saul A Teukolsky, William T Vetterling & Brian P Flannery. *Numerical Recipes in C.* Second ed., Cambridge University Press, 1992.

Price, Jason E. "Representation and Parameterisation Issues in Genetic Algorithms." PhD thesis, Aston University, 1996.

Rabenhorst, David A, Edward J Farrell, David H Jameson, Thomas D Linton & Jack A Mandelman. "Complementary Visualization and Sonification of Multi-dimensional Data." In *SPIE Vol. 1259 Extracting Meaning from Complex Data: Processing, Display, Interaction,* 1990: 147–153.

Rao, Ramana & Stuart K Card. "The Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus+Context Visualisation for Tabular Information." In *Proceedings of CHI '94*. ACM, 1994.

Rao, Ramana & Stuart K Card. "Exploring Large Tables with the Table Lens." In *Proceedings of CHI '95*. ACM, 1995.

Rheingans, Penny. "Color, Change and Control for Quantitative Data Display." In *Visualization '92* (Boston, Massachusetts), Arie E Kaufman & Gregory M Nielson ed. IEEE Computer Society Press, 1992: 252–259.

Rheingold, Howard. *Virtual Reality*. London: Secker & Warburg, 1991.

Robertson, George G, Stuart K Card & Jock D Mackinlay. "The Cognitive Coprocessor Architecture for Interactive User Interfaces." In *Proceedings of the ACM SIGRAPH Symposium on User Interface Software and Technology*. ACM, 1989: 10–18.

Robertson, George G, Jock D Mackinlay & Stuart K Card. "Cone Trees: Animated 3-D Visualisations of Hierarchical Information." In *Proceedings of CHI '91*. ACM, 1991A: 189–194.

Robertson, George G, Jock D Mackinlay & Stuart K Card. "Information Visualisation using 3-D Interactive Animation." In *Proceedings of CHI '91*. ACM, 1991B: 461–462.

Sammon, John W. "A Nonlinear Mapping for Data Structure Analysis." *IEEE Transactions on Computers* C-18, no. 5 (1969): 401–409.

Sarkar, Manojit & Marc H Brown. "Graphical Fisheye Views of Graphs." In *Proceedings of CHI '92* (Monterey, California), Penny Bauersfeld, John Bennett & Gene Lynch ed. Addison Wesley, 1992.

Sarkar, Manojit & Steven P Reiss. *Manipulating Screen Space with StretchTools: Visualizing Large Structure on Small Screens*. Department of Computer Science, Brown University (Ref: CS-92-42), 1992.

SAS Institute Inc. "The FASTCLUS Procedure." In *SAS/STAT User's Guide, Release 6.03 Edition*. Cary, NC: SAS Institute Inc, 1988: 493–518.

Scott, David W. "On estimation and visualisation of higher dimensional surfaces." In *Computing and Graphics in Statistics*, ed. Andreas Buja & Paul A Tukey. New York: Springer-Verlag, 1991.

Scott, David W. *Multivariate Density Estimation: Theory, Practice, and Visualization.* New York: John Wiley & Sons, 1992.

Sherman, Barrie & Phil Judkins. *Glimpses of Heaven, Visions of Hell.* London: Hodder & Stoughton, 1992.

Silverman, Bernard W. *Density Estimation for Statistics and Data Analysis.* London: Chapman & Hall, 1986.

Stephenson, Neal. *Snow Crash.* New York: Bantam, 1992.

Stone, Maureen C, Ken Fishkin & Eric A Bier. "The Movable Filter as a User Interface Tool." In *Proceedings of CHI '94* (New York). ACM, 1994.

Tattersall, G D & P R Limb. "Visualisation techniques for data mining." *BT Technology Journal* 12, no. 4 (1994).

Titterington, D M, A F M Smith & E U Makov. *Statistical Analysis of Finite Mixture Distributions.* Chichester: John Wiley & Sons, 1985.

Tufte, Edward R. *The Visual Display of Quantitative Information.* Cheshire, Conn.: Graphics Press, 1983.

Tufte, Edward R. *Envisioning Information.* Cheshire, Conn.: Graphics Press, 1990.

Tversky, Oren J, Scott S Snibbe & Robert Zeleznik. *Cone Trees in the UGA Graphics System: Suggestions of a more Robust Visualisation Tool.* Department of Computer Science, Brown University (Ref: CS-93-07), 1993.

Tweedie, Lisa. *Prosection.* Personal communications, 1995.

Tweedie, Lisa, Robert Spence, Huw Dawkes & Hua Su. "Externalising Abstract Mathematical Models." In *Proceedings of CHI '96* (submitted), 1996.

Waldrop, M Mitchell. "Learning to drink from a fire hose." *Science*, 11 May 1990, 674–675.

Walker, Graham R, Paul A Rea, Steven Whalley, Mike Hinds & Nicholas J Kings. "Visualisation of telecommunications network data." *BT Technology Journal* 11, no. 4 (1993): 54–63.

Ware, Colin & William Knight. "Orderable Dimensions of Visual Texture for Data Display: Orientation, Size and Contrast." In *CHI '92* (Monterey, California), Penny Bauersfeld, John Bennett & Gene Lynch ed., Addison Wesley, 1992.

Ware, Colin & William Knight. "Using Visual Texture for Information Display." *ACM Transactions on Graphics* 14, no. 1 (1995): 3–20.

Watt, Alan. *Fundamentals of Three-dimensional Computer Graphics.* Wokingham: Addison-Wesley, 1989.

Appendix: Database Details

A.1 Introduction

Three databases were obtained to develop and test the visualisation tools described in this thesis. This chapter gives details of the names, range of values and (where possible) meanings of all the fields in the three databases.

A.2 Mail Database

For confidentiality, the mail database was generated by simulations based on a real customer database from a financial institution. The list below describes its fields.

A.2.1 Field details

Age	customer's age	Integer (0–80). 22% unknown (age=0)
Ac_Turn	account turnover	Integer (0–1930)
No_Wdraw	number of withdrawals	Integer (0–10)
Minbal	minimum balance	Integer (-2000–1501). 10% have Minbal=-2000
Maxbal	maximum balance	Integer (0–1786)
Ac_Age	account age	Integer (0–20)
No_pre	pre-payments[1]	Integer (0–8)
Ccard	credit card	Binary (1=has card)
Dcard	debit card	Binary (1=has card)
Mortgage	type of mortgage held	Categorical (C, M, N, Unknown)
Cont_ins	contents insurance	Binary (1=has contents insurance)
Buil_ins	buildings insurance	Binary (1=has buildings insurance)
Life_ins	life insurance	Binary (1=has life insurance)
Pension	pension	Binary (1=has pension)
Pers_loan	personal loan	Binary (1=has personal loan)
Acorn	acorn classification[2]	Categorical (A–K and Unknown). 0.4% unknown
Sex	customer's sex	Categorical (Female, Male, Unknown). 10% unknown
Marital	marital status	Categorical (Married, Other, Single, 15% Unknown)
Home	home status	Categorical (Outright, Paying, Rented, 27% Unknown)
Response	response to mail shot	Binary (1=positive response)

[1] Number of direct debits, standing orders etc.

[2] The Acorn classification is based on data from the last census

A.3 Finance Database

As has been lamented many times throughout this thesis, the meanings of the fields in the finance database were not revealed by Recognition Systems for reasons of client confidentiality. The list below therefore shows only the field names, types and ranges.

A.3.1 Field details

t1	Categorical (A–I)
t2	Categorical (A–F, Unknown). 12% unknown
t3	Categorical (D, E, S, Unknown). 13% unknown
q1	Categorical (Yes, No, Unknown). 15% unknown
q2	Categorical (Yes, No, Unknown). 21% unknown
q3	Categorical (Yes, No, Unknown). 27% unknown
q4	Categorical (Yes, No, Unknown). 2% unknown
q5	Categorical (Yes, No, Unknown). 2% unknown
c1	Continuous (473.27–507.86)
c2	Continuous (-10.49–32.40)
c3	Continuous (219.09–252.47)
b1	Binary
b2	Binary
b3	Binary
b4	Binary
b5	Binary
b6	Binary
c4	Continuous (2146.68–2209.51)
q6	Categorical (Yes, No, Unknown). 7% unknown
c5	Continuous (-356.47–195231.24)
c6	Continuous (-5.55–149.81)
response	Binary

A.4 RAE Database

A.4.1 References

All the data was indexed by *institution* and *unit of assessment*. There were 177 institutions and 72 units of assessment, as detailed at the end of this chapter.

A.4.2 Pre-processing

The data was supplied in seven separate files, containing a large number of fields, several of which were found to be redundant. Also, certain figures were quoted for several years (student numbers, numbers and values of grants for 1988–91, publications produced for 1988–92, and cited publications for 1988–93). The means of all these figures across the years was used, to reduce the number of fields.

A.4.3 Field details

The value -1 in most fields signifies that the real value is unknown, because the department concerned did not submit the required information. Many of the staffing values are given as full-time equivalents (FTE); and the non-integer values in most fields are due to the averaging described earlier.

#inst	Integer (100–514)	Institution number
#uofa	Integer (1–72)	Unit of assessment code
sel_staf	Continuous (0.4–178.8)	Number of category A selected staff
non_staf	Continuous (0–183.3)	Number of staff not selected
ra_postd	Continuous (0–129.2)	Number of postdoctoral research assistants (FTE)
ra_postg	Continuous (0–216)	Number of postgraduate research assistants (FTE)
tech	Continuous (0–143.5)	Number of technicians (FTE)
sci_off	Continuous (0–62)	Number of scientific officers (FTE)
exp_off	Continuous (0–23)	Number of experimental officers (FTE)
other	Continuous (0–43)	Number of other staff (FTE)
general	Integer (0–224)	Number of staff funded from general income
other_inc	Integer (0–229)	Number of staff funded from other income
inpost	Integer (0–216)	Number of staff in post throughout assessment period
publicns	Integer (0–5053)	Number of publications
ras	Continuous (0–418.7)	Number of research assistants
students	Continuous (0–510)	Number of postgraduate research students
cited_1	Integer (-1–68)	Number of cited authored books
cited_2	Integer (-1–35)	Number of cited edited books
cited_3	Integer (-1–158)	Number of cited short works
cited_4	Integer (-1–66)	Number of cited refereed conferences
cited_5	Integer (-1–22)	Number of cited other conferences
cited_6	Integer (-1–13)	Number of cited editorships
cited_7	Integer (-1–415)	Number of cited articles for academic journals
cited_8	Integer (-1–159)	Number of cited articles for profesional journals
cited_9	Integer (-1–5)	Number of cited articles for popuar journals
cited_10	Integer (-1–8)	Number of cited reviews of academic books
cited_11	Integer (-1–41)	Number of cited other publications
cited_12	Integer (-1–220)	Number of cited miscellaneous publications
cited_apd	Integer (-1–21)	Number of cited pubs classified as applied research
n_ft_res	Continuous (-1–544.5)	Number in full-time research (average)
n_pt_res	Continuous (-1–292.5)	Number in part-time research (average)
n_doctoral	Continuous (-1–59.25)	Number of doctorates per year (average)
n_masters	Continuous (-1–244.25)	Number of masters per year (average
n_fte_postgrads	Continuous (-1–548.12)	Number of postgraduate students (FTE, average)
stud_a	Continuous (-1–199)	Number of studentships from ABRC research councils
stud_b	Continuous (-1–123)	Number of studentships from UK-based charities
stud_c	Continuous (-1–63)	Number of studentships from UK central government
stud_d	Continuous (-1–84)	Number of studentships from UK local government
stud_e	Continuous (-1–15)	Number of studentships from UK public corporations

stud_f	Continuous (-1–125)	Number of studentships from UK industry & commerce
stud_g	Continuous (-1–59)	Number of studentships from UK health & HAS
stud_h	Continuous (-1–133)	Number of studentships from other overseas
stud_oth	Continuous (-1–104)	Number of studentships from others
num_grant_a	Continuous (-1–273.2)	Number of grants from ABRC research councils *et al*
num_grant_b	Continuous (-1–386)	Number of grants from UK-based charities
num_grant_c	Continuous (-1–103)	Number of grants from UK central government
num_grant_d	Continuous (-1–83)	Number of grants from UK local government
num_grant_e	Continuous (-1–35)	Number of grants from UK public corporations
num_grant_f	Continuous (-1–1002)	Number of grants from UK industry & commerce
num_grant_g	Continuous (-1–104.6)	Number of grants from UK health & HAS
num_grant_h	Continuous (-1–44)	Number of grants from EC
num_grant_i	Continuous (-1–61)	Number of grants from other overseas
num_grant_j	Continuous (-1–15)	Number of grants from PCFC/NAB initiatives
num_grant_k	Continuous (-1–17.8)	Number of grants: teaching company schemes – RCS
num_grant_l	Continuous (-1–17.8)	Number of grants: teaching company schemes – comp
num_grant_oth	Continuous (-1–109)	Number of grants from others
val_grant_a	Integer (-1–2.14e7)	Value of grants from ABRC research councils *et al*
val_grant_b	Integer (-1–2.87e7)	Value of grants from UK-based charities
val_grant_c	Integer (-1–1.51e7)	Value of grants from UK central government
val_grant_d	Integer (-1–842105)	Value of grants from UK local government
val_grant_e	Integer (-1–2.03e6)	Value of grants from UK public corporations
val_grant_f	Integer (-1–1.31e7)	Value of grants from UK industry & commerce
val_grant_g	Integer (-1–3.49e6)	Value of grants from UK health & HAS
val_grant_h	Integer (-1–4.81e6)	Value of grants from EC
val_grant_i	Integer (-1–8.08e6)	Value of grants from other overseas
val_grant_j	Integer (-1–1.14e6)	Value of grants from PCFC/NAB initiatives
val_grant_k	Integer (-1–994704)	Value of grants: teaching company schemes – RCS
val_grant_l	Integer (-1–1.12e6)	Value of grants: teaching company schemes – comp
val_grant_oth	Integer (-1–5e6)	Value of grants from others
rc_usage	Integer (-1–15000)	Usage of research council facilities (£)
pub_1	Continuous (-1–146)	Number of published authored books
pub_2	Continuous (-1–331.2)	Number of published edited books
pub_3	Continuous (-1–1751.9)	Number of published short works
pub_4	Continuous (-1–917.9)	Number of published refereed conferences
pub_5	Continuous (-1–441)	Number of published other conferences
pub_6	Continuous (-1–171.2)	Number of published editorships
pub_7	Continuous (-1–2192.3)	Number of published articles for academic journals
pub_8	Continuous (-1–407.9)	Number of published articles for professional journals
pub_9	Continuous (-1–178)	Number of published articles for popular journals
pub_10	Continuous (-1–771)	Number of published reviews of academic books
pub_11	Continuous (-1–380.7)	Number of published other publications
pub_12	Continuous (-1–1058)	Number of published miscellaneous publications
staff_pub	Continuous (-1–249)	Number of staff producing publications (average)
rating	Integer (1–5)	Research rating awarded

A.4.4 Institution codes

#inst	Institution	#inst	Institution
100	Anglia Polytechnic University	101	Aston University
102	Bath College	103	University of Bath
105	University of Birmingham	107	Bolton Institute
108	Bournemouth University	109	University of Bradford
111	University of Brighton	112	University of the West of England
113	University of Bristol	114	Brunel University
115	Buckinghamshire College	116	Camborne School of Mines
117	University of Cambridge	118	Central School of Speech & Drama
120	Cheltenham & Gloucester College	121	Chester College
122	Canterbury Christ Church College	123	City of London Polytechnic
124	City University	125	College of Ripon & York St John
126	College of St Mark & St John	127	Coventry University
128	Cranfield Institute of Technology	129	Crewe & Alsager College
130	Dartington College of Arts	131	University of Derby
132	University of Durham	133	University of East Anglia
135	University of Essex	136	Universiry of Exeter
138	Harper Adams Agricultural College	139	University of Hertfordshire
140	Homerton College	141	University of Huddersfield
142	University of Hull	143	University of Humberside
145	University of Keele	147	University of Kent at Canterbury
149	Kingston University	150	University of Central Lancashire
151	University of Lancaster	152	La Sainte Union College
153	Leeds Metropolitan University	154	University of Leeds
155	De Montfort University	156	University of Leicester
158	Liverpool John Moores University	159	University of Liverpool
160	London Business School	162	Birkbeck College
164	Charing Cross & Westminster Med	165	Goldsmiths' College
166	Imperial College	167	Institute of Education
168	King's College London	169	London Sch of Economics & Pol Sci
170	London Hospital Medical College	171	London Sch of Hygiene & Trop Med
172	Queen Mary & Westfield College	173	Royal Free Hospital Sch of Medicine
174	Royal Holloway & Bedford New Coll	175	Royal Postgraduate Medical School
176	Royal Veterinary College	177	St Bartholomew's Hospital Med Coll
178	St George's Hospital Medical School	179	School of Oriental & African Studies
180	School of Pharmacy	181	United Medical & Dental Schools
182	University College London	183	Wye College
184	British Institute in Paris	186	Courtauld Institute of Art
187	Institute of Advance Legal Studies	188	Institute of Classical Studies
189	Institute of Commonwealth Studies	190	Institue of Germanic Studies
191	Institute of Historical Research	193	Institute of Romance Studies
194	School of Slavonic & E Euro Studies	195	Warburg Institute
196	University Marine Biol Stn Millport	198	Loughborough Univ of Technology
199	University of Luton	200	Manchester Business School
201	Manchester Metropolitan University	202	University of Manchester

203	UMIST	204	Middlesex University
205	Nene College	206	Univ of Northumbria at Newcastle
207	University of Newcastle upon Tyne	210	Nottingham Trent University
211	University of Nottingham	212	Open University
213	Oxford Brookes University	214	University of Oxford
215	University of Westminster	216	University of East London
217	University of North London	218	Thames Valley University
219	University of Plymouth	220	University of Portsmouth
222	University of Reading	223	Roehampton Institute
226	Royal College of Art	229	St Mary's College
230	S. Martin's College	231	Salford College of Technology
232	University of Salford	233	Sheffield Hallam University
234	University of Sheffield	235	South Bank University
236	Southampton Institute	237	University of Southampton
238	Staffordshire University	239	University of Sunderland
240	University of Surrey	241	University of Sussex
242	University of Teesside	243	University of Greenwich
244	The London Institute	245	Trinity & All Saints College
247	University of Warwick	248	Westminster College
249	West London Institute	250	West Surrey College of Art & Design
252	Winchester School of Art	253	University of Wolverhampton
254	Worcester College	255	University of York
256	BPMF: Institute of Neurology	257	BPMF: Institute of Child Health
258	BPMF: Natnl Heart & Lung Institute	259	BPMF: Institute of Opthalmology
260	BPMF:Institute of Psychiatry	261	BPMF: Institute of Dental Surgery
262	BPMF: Institute of Cancer Research	263	Institute of Zoology
264	Wimbledon School of Art	300	Queen's University of Belfast
302	Stranmillis College of Education	303	University of Ulster
304	Armagh Observatory	400	University of Aberdeen
401	Craigie College of Technology	402	Duncan of Jordanstone College of Art
403	Dundee Institute of Technology	404	University of Dundee
405	Edinburgh College of Art	406	University of Edinburgh
407	Glasgow Polytechnic	408	Glasgow School of Art
409	University of Glasgow	410	Heriot-Watt University
411	Jordanhill College of Education	412	Moray House College of Education
413	Napier University	414	Northern College of Education
415	University of Paisley	416	Queen Margaret College
417	Queen's College Glasgow	418	Robert Gordon University
421	University of St Andrews	424	University of Stirling
425	University of Strathclyde	503	North East Wales Institute
504	University of Glamorgan	505	Trinity College Carmarthen
506	University Coll of Wales Aberystwyth	507	University Coll of North Wales Bangor
508	University of Wales Coll of Cardiff	509	University College of Swansea
510	University of Wales Coll of Medicine	511	St David's University College
514	Swansea Institute		

A.4.5 Unit of assessment codes

#uofa	Unit of Assessment	#uofa	Unit of Assessment
1	Clinical Lab Sciences	2	Community Clinical
3	Hospital Clinical	4	Clinical Dentistry
5	Pre-clinical Studies	6	Anatomy
7	Physiology	8	Pharmacology
9	Pharmacy	10	Nursing
11	Other Medicine	12	Biochemistry
13	Psychology	14	Biological Sciences
15	Genetics	16	Microbiology
17	Agriculture	18	Food Science & Tech
19	Veterinary Science	20	Chemistry
21	Physics	22	Earth Sciences
23	Environmental Studies	24	Pure Mathematics
25	Applied Mathematics	26	Stats & Operational
27	Computer Science	28	General Engineering
29	Chemical Engineering	30	Civil Engineering
31	Elec & Elec Eng	32	Mech, Aero & Manuf Eng
33	Mineral & Mining Eng	34	Metallurgy & Materials
35	Built Environment	36	Town & Country Planning
37	Geography	38	Law
39	Anthropology	40	Economic & Social Hist
41	Economics etc	42	Politics & Intl Studies
43	Social Policy & Admin	44	Social Work
45	Sociology	46	Business & Management
47	Accountancy	48	American Studies
49	Middle E & African	50	E & S Asian Studies
51	English Lang & Lit	52	European Studies
53	Celtic Studies	54	French
55	German & Related	56	Italian
57	Russian	58	Spanish
59	Linguistics	60	Classics & Ancient Hist
61	Archaeology	62	History
63	History of Art etc	64	Library & Info Mgt
65	Philosophy	66	Theology, Diviniry +RS
67	Art & Design	68	Communication & Media
69	Drama, Dance & Perf	70	Music
71	Education	72	PE & Sports Sciences